

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
*Кафедра автоматизованих систем обробки інформації і управління*

До захисту допущено:

В.о. завідувача кафедри

\_\_\_\_\_ *Олександр ПАВЛОВ*  
(підпис) (вл.ім'я, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2020 р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інформаційні управляючі  
системи та технології»  
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: «Інформаційна система пошуку ресторанів »

**Виконав**

студент IV курсу, групи ІС-62

\_\_\_\_\_ *Федько Микола Романович*

(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Керівник**

\_\_\_\_\_ *ст. викл. Ковтунець Олесь Володимирович*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Консультант з  
графічної  
документації**

\_\_\_\_\_ *к.т.н., доц. Новінський Валерій Петрович*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

**Рецензент**

\_\_\_\_\_ *к.т.н., доц. Пасько Віктор Петрович*

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент

\_\_\_\_\_ (підпис)

Київ – 2020 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет (інститут) інформатики та обчислювальної техніки  
(повна назва)

Кафедра автоматизованих систем обробки інформації та управління  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ  
(підпис) (вл. ім'я, прізвище)

“ ” 2020 р.

**ЗАВДАННЯ  
на дипломний проєкт студенту**

Федьку Миколі Романовичу  
(прізвище, ім'я, по батькові)

1. Тема проєкту «Інформаційна система пошуку ресторанів»

керівник проєкту Ковтунець Олесь Володимирович, ст. вик.  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7” травня 2020 р. № 1081-с

2. Термін подання студентом проєкту “01” червня 2020 року

3. Вихідні дані до проєкту

*Технічне завдання*

4. Зміст пояснювальної записки

*1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі*

*2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних*

*3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання*

*4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів*

*5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту*

5. Перелік графічного матеріалу

*1. Схема структурна варіантів використання*

3. *Схема бази даних*

4. *Схема структурна класів програмного забезпечення*

5. *Схема структурна послідовності*

6. *Схема структурна компонентів програмного забезпечення*

7. *Креслення екранних форм*

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання *«13» квітня 2020 року*

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	10.03.2020 р.	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	15.03.2020 р.	
3.	<i>Постановка та формалізація задачі</i>	28.03.2020 р.	
4.	<i>Розробка інформаційного забезпечення</i>	07.05.2020 р.	
5.	<i>Алгоритмізація задачі</i>	22.03.2020 р.	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	30.03.2020 р.	
7.	<i>Розробка програмного забезпечення</i>	10.04.2020 р.	
8.	<i>Налагодження програми</i>	17.04.2020 р.	
9.	<i>Виконання графічних документів</i>	23.04.2020 р.	
10.	<i>Оформлення пояснювальної записки</i>	01.05.2020 р.	
11.	<i>Подання ДП на попередній захист</i>	15.05.2020 р.	
12.	<i>Подання ДП на основний захист</i>	01.06.2020 р.	
13.	<i>Подання ДП рецензенту</i>	02.06.2020 р.	

Студент

Микола ФЕДЬКО

Керівник

Олесь КОВТУНЕЦЬ

[illegible]

# **Пояснювальна записка до дипломного проєкту**

на тему: Інформаційна система пошуку ресторанів

---

---

---

Київ – 2020 року

## АНОТАЦІЯ

**Структура та обсяг роботи.** Пояснювальна записка дипломного проєкту складається з шести розділів, містить 92 сторінки, 25 рисунків, 36 таблиць, 1 додаток, 18 джерел. Дипломний проєкт присвячений розробці інформаційної системи пошуку ресторанів з ціллю скоротити час на пошуки ресторану, шляхом надання релевантних рекомендацій. У розділі загальних положень було наведено словесний опис предметного середовища. Визначено призначення, цілі та задачі розробки застосування.

У розділі інформаційного забезпечення було визначено вхідні та вихідні дані до комплексу задач, була розроблена структура бази даних для збереження даних, яка відповідає поставленим цілям проєкту.

Розділ математичного забезпечення присвячений опису та обґрунтуванню методів розв’язання поставлених задач.

У розділі програмного забезпечення описані основні засоби розробки системи, висунуті вимоги до технічного забезпечення, обрано та обґрунтовано архітектуру програмного забезпечення.

У технологічному розділі наведена інструкція для користувача та проведено тестування системи.

РЕЛЕВАНТНА ПРОПОЗИЦІЯ, ПУБЛІКАЦІЯ, РЕСТОРАН,  
СТРАВА, КОЛАБОРАТИВНА ФІЛЬТРАЦІЯ, ПОШУК.

					ДП 6225.00.000 ПЗ			
		Прізвище	Підпис	Дата				
Розроб.	Федько М.Р.					Лім.	Лист	Листів
Перевірив.	Ковтунець О.В.						2	
					Інформаційна система пошуку ресторанів	КПІ ім. Ігоря Сікорського		
Н. кон..	Новінський В.П.					Каф. АСОІУ		
Затв.	Ковтунець О.В.					Гр. ІС-62		
								1

## ABSTRACT

**Structure and scope of work.** The explanatory note of the diploma project consists of six sections, contains 92 pages, 25 pictures, 36 tables, 1 application, 18 sources.

The diploma project is devoted to the development of an information system for restaurants finding in order to reduce the time for searching a restaurant, by providing relevant recommendations.

In the section of general provisions was given a description of the objective environment. Defined purposes, aims and objectives of system development. Also described basic processes of business and functional model.

In the section of information support the input and output data to a complex of tasks were defined, the structure of a database for data storage which corresponds to the set purposes of the project was developed.

The section of mathematical support is devoted to the description and substantiation of methods of solving the set tasks.

The software section describes the main tools of system development, the requirements for hardware, selected and justified the software architecture.

The technological section provides user manual. Passed the test match system functions requirements specification.

RELEVANT OFFER, PUBLICATION, RESTAURANT, DISH, COLLABORATIVE FILTRATION, SEARCH.

						Арк.
						2
ЗМН.	Арк.	№ докум.	Підпис	Дата		

## ЗМІСТ

ВСТУП.....	5
<b>1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....</b>	<b>6</b>
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	6
1.1.1 Опис процесу діяльності.....	6
1.1.2 Опис функціональної моделі.....	7
1.2 ОГЛЯД НАЯВНИХ АНАЛОГІВ.....	8
1.3 ПОСТАНОВКА ЗАДАЧІ.....	8
1.3.1 Призначення розробки .....	8
1.3.2 Цілі та задачі розробки.....	9
Висновок до розділу.....	9
<b>2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>10</b>
2.1 ВХІДНІ ДАНІ .....	10
2.2 ВИХІДНІ ДАНІ.....	10
2.3 ОПИС СТРУКТУРИ БАЗИ ДАНИХ.....	11
2.4 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ.....	12
Висновок до розділу.....	15
<b>3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>16</b>
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ.....	16
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ.....	17
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ .....	18
3.4 ОПИС МЕТОДІВ РОЗВ’ЯЗАННЯ .....	18
Висновок до розділу.....	19
<b>4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>20</b>
4.1 ЗАСОБИ РОЗРОБКИ.....	20
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ .....	22
4.2.1 Загальні вимоги .....	22
4.3 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
4.3.1 Діаграма класів .....	23
4.3.2 Діаграма послідовності.....	24
4.3.3 Діаграма компонентів .....	24



4.3.4	Специфікація функцій.....	24
	Висновок до розділу.....	26
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	27
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА .....	27
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ .....	39
5.2.1	Мета випробувань .....	39
5.2.2	Загальні положення .....	40
5.2.3	Результати випробувань .....	40
	Висновок до розділу.....	59
	ЗАГАЛЬНІ ВИСНОВКИ.....	60
	ПЕРЕЛІК ПОСИЛАНЬ .....	61
	ДОДАТОК А .....	63

## ВСТУП

У сучасному світі більшість людей схиляється до того, щоб ефективно розпоряджатися власним часом. Одним з методів економії часу може стати харчування в ресторанах або ж замовлення їжі з доставкою. Багатьох людей, що живуть в великих містах затягує рутинна і вони часто стають перед важким вибором: де пообідати, у якому ресторані провести бізнес-зустріч або ж у якому закладі провести незабутнє побачення. Для таких людей рекомендація найкращого ресторану буде надзвичайно цінною, оскільки збереже час та сили на пошуки. Саме тому виникла ідея створити сервіс, який допомагав би людям отримувати релевантні рекомендації щодо ресторанів міста.

Дипломний проєкт присвячений розробці інформаційної системи пошуку ресторанів з ціллю скоротити час на пошуки ресторану, шляхом надання релевантних рекомендацій.

**Практичне значення одержаних результатів.** Розроблено алгоритм для формування персональних рекомендацій.

						Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Опис предметного середовища

Розроблюваний програмний продукт являє собою веб платформу з рейтингами ресторанів міста та рекомендаціями . Ідея платформи полягає в тому щоб формувати рекомендації для людей, який ресторан відвідати — на основі рейтингів ресторанів, вподобань користувача та схожих за інтересами людей.

На платформі представлені ресторани міста з різними стравами. Також у є користувачі, які відвідують певний ресторан, замовляють страву і потім виставляють оцінку страві та ресторану. Система знаходить користувачів зі схожими вподобаннями і формуємо "персональні" рекомендації для них.

Система вирішує проблеми як відвідувачів, так і власників ресторанів. Дана платформа спрощує процес пошуку ресторану, надаючи персональні рекомендації для користувача. Ресторани в свою чергу отримують додаткове джерело трафіку, також мають можливість покращувати рівень обслуговування, проаналізувавши відгуки користувачів платформи.

#### 1.1.1 Опис процесу діяльності

Інформаційна система пошуку ресторанів являє собою веб–платформу, яка передбачає користування для 3 типів користувачів: адміністратора системи, зареєстрованого користувача та не зареєстрованого користувача, тобто гостя. Різниця полягає у функціональних можливостях та рівнях доступу для користувачів.

Діаграма варіантів використання розробленої платформи знаходиться у додатку з графічними матеріалами. У даній діаграмі наглядним чином показано які дії може виконувати кожен з акторів.

						Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.1.2 Опис функціональної моделі

Акторами платформи є:

- CUSTOMER – зареєстрований користувач системи;
- GUEST – гість системи
- ADMIN – адміністратор системи.

Функції, які будуть виконуватись акторами:

#### CUSTOMER:

- Додавання / видалення уподобань та переваг;
- Перегляд рейтингу ресторанів;
- Перегляд персональних рекомендацій;
- Публікація оцінки та відгуку для страви та ресторану;
- Пошук страви / ресторану / кухні у системі;
- Фільтрація / сортування рейтингів, рекомендацій та результатів пошуку за ціною, оцінкою, назвою чи місцезнаходженням.

#### GUEST:

- Реєстрація в системі;
- Перегляд рейтингу ресторанів;
- Пошук страви / ресторану / кухні у системі;
- Фільтрація / сортування рейтингів, рекомендацій та результатів пошуку за ціною, оцінкою, назвою чи місцезнаходженням.

#### ADMIN:

- Перегляд списку зареєстрованих користувачів;
- Надання доступу адміністратора для користувача;
- Видалення користувача;
- Додавання / видалення / редагування даних про страву, кухню чи ресторан.

## 1.2 Огляд наявних аналогів

Сьогодні на ринку існує кілька інформаційних продуктів, орієнтованих на формування рейтингів організацій зі сфери послуг, місць відпочинку, у тому числі й ресторанів.

Подібні платформи як TripAdvisor або Yelp збирають відгуки користувачів та формують рейтинги для ресторанів, готелів, авіакомпаній та інших типів компаній зі сфери послуг.

TripAdvisor — американський веб-сайт, який дозволяє користувачам планувати майбутні подорожі до будь-яких країн світу. Послуги сайту безкоштовні. За даними офіційного веб-сайту компанії, TripAdvisor щомісяця обслуговує 315 мільйонів унікальних відвідувачів, більш ніж 70 мільйонів зареєстрованих користувачів, які залишили більше ніж 200 мільйонів відгуків.

Yelp — веб-сайт для пошуку на місцевому ринку послуг, наприклад ресторанів або перукарень, з можливістю додавати та переглядати рейтинги та огляди цих послуг. Для популярних бізнесів є сотні оглядів. Для оглядачів на сайті передбачені елементи соціальної мережі.

## 1.3 Постановка задачі

### 1.3.1 Призначення розробки

Даний програмний продукт призначений для того, щоб спростити пошук ресторану для користувача, шляхом формування персональних рекомендацій. Система повинна аналізувати рейтинги ресторанів, уподобання, оцінки та досвід користувача, щоб сформуванати для нього якісні пропозиції.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

### 1.3.2 Цілі та задачі розробки

Ціль створення системи:

- Спростити життя користувачів та скоротити час на пошуки ресторану, шляхом надання релевантних рекомендацій;

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- Розробити алгоритм для формування рекомендацій;
- Розробити алгоритм для формування персоналізованих рекомендацій;
- Розробити зручний та зрозумілий інтерфейс для користування системою;

### Висновок до розділу

На даному етапі дипломного проекту були визначені мета та цілі розробки системи, предметне середовище. Також було проведене дослідження існуючих аналогів програмного продукту. Сформований опис функціональної моделі, а саме: наведена коротка характеристика дійових осіб, тобто акторів системи та їх можливостей. Визначені основні вимоги до розроблюваної системи та задачі, які потрібно реалізувати в рамках дипломного проекту.

						Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

## 2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

### 2.1 Вхідні дані

Вхідні дані можуть надходити в систему від кількох джерел, а саме від адміністратора, гостя платформи, або повноцінного користувача.

Адміністратор може вводити наступні дані:

- інформацію про ресторан, а саме: назву ресторану, місцезнаходження, контактні дані, середній чек та зображення ;
- інформацію про страву: назву страви, опис, ціну та зображення страви;
- інформацію про кухню: назву кухні та зображення.

Гість може зареєструватись у системі та стати повноцінним користувачем, для цього він має ввести:

- ім'я, пошту та пароль, необхідні для реєстрації особистого кабінету;

Користувач має змогу ввести:

- оцінку та відгук про відвіданий ресторан;
- страву або кухню (зі списку тих, що розміщених на платформі) у список власних уподобань.

### 2.2 Вихідні дані

Вихідними даними системи є сформовані програмно списки з рейтингами ресторанів та списки персональних рекомендацій для користувача. Персональна рекомендація має вигляд публікації, яка містить інформацію про страву, кухню та ресторан.

Приклад вихідних даних зображений на рисунку 2.1

						Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

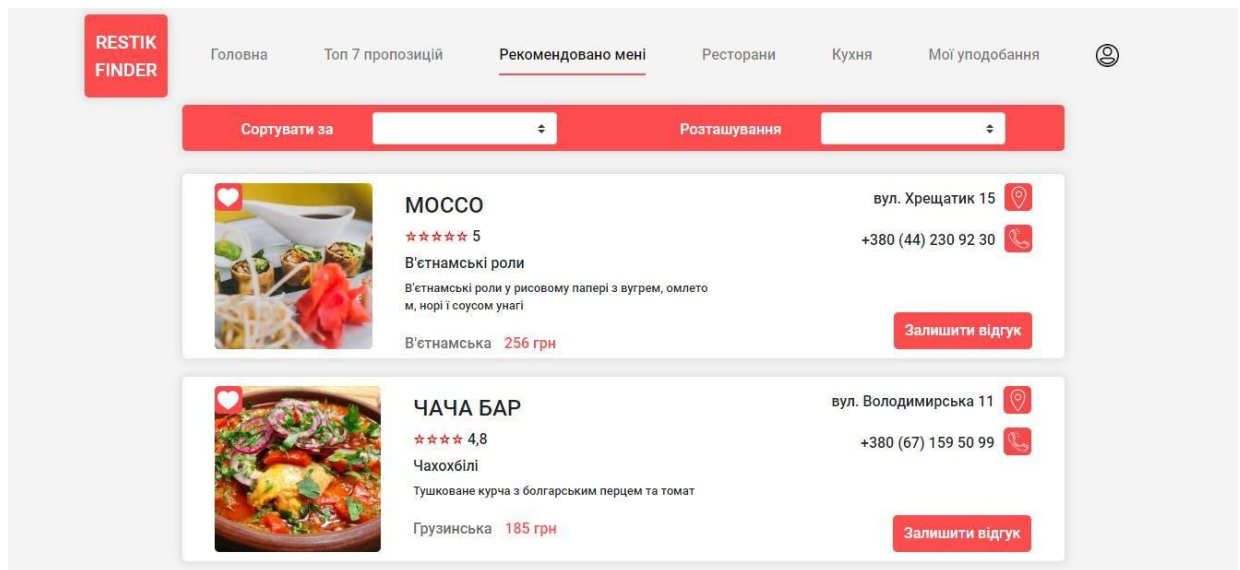


Рисунок 2.1 — Сформована персональна рекомендація

## 2.3 Опис структури бази даних

Для зберігання даних була використана база даних з такими таблицями: User, Restaurant, Cuisine, Dish, Preference, Recommendation, Review. Схема структурна бази даних системи зображена на рисунку 2.2.

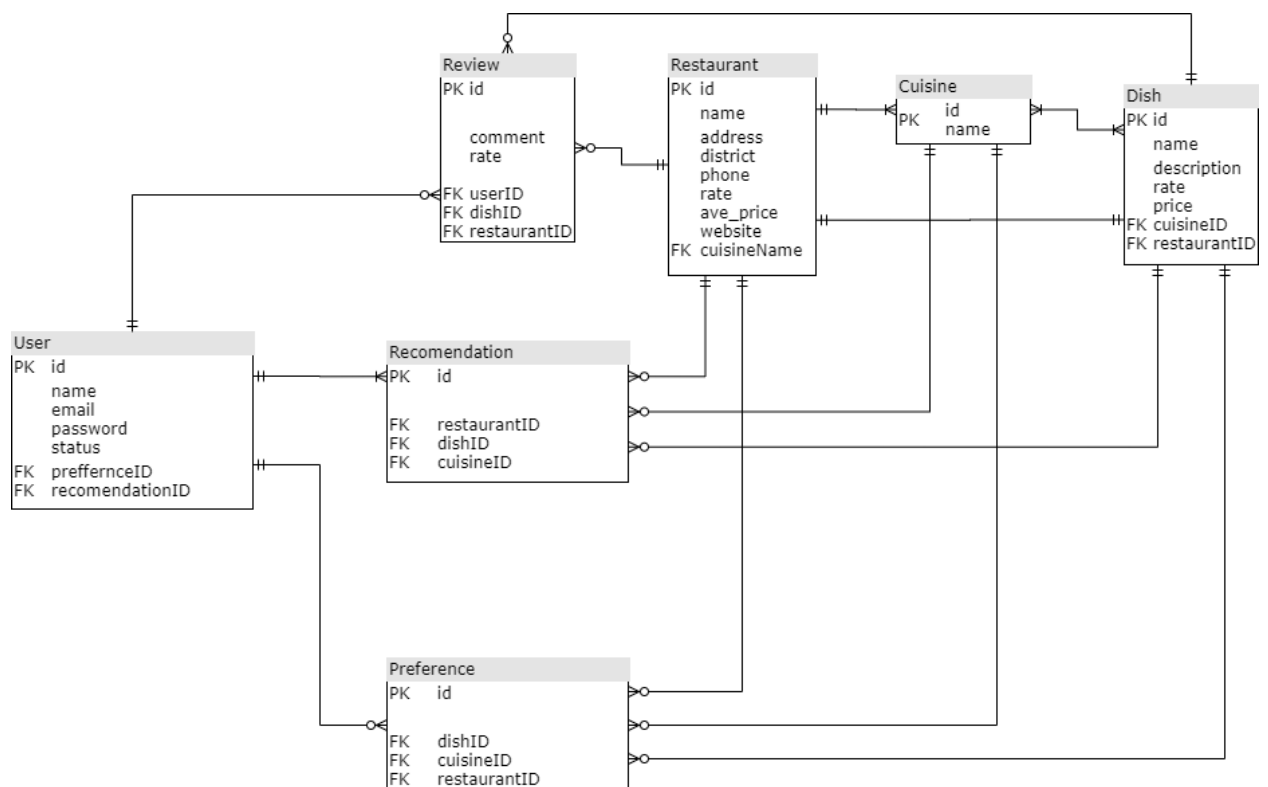


Рисунок 2.2 — Схема структурна бази даних системи у вигляді ERD діаграми



## 2.4 Структура масивів інформації

У таблиці 2.1 розміщений безпосередньо опис вмісту таблиць розроблених для бази даних.

Таблиця 2.1 — Опис вмісту таблиць бази даних

Назва таблиці	Назва стовпця	Тип даних	Опис даних
User	id	integer	Код користувача, який ідентифікує його в БД
	status	Boolean	Рівень доступу до сервісу (користувач / адміністратор)
	name	varchar	Ім'я користувача
	email	varchar	Пошта користувача
	password	varchar	Пароль користувача
	preferenceID	integer	Код уподобання користувача
	recommendationID	integer	Код рекомендації користувача
Restaurant	id	integer	Код ресторану, який ідентифікує його в БД
	name	varchar	Назва ресторану
	address	varchar	Адреса ресторану
	phone	varchar	Контактний номер телефону
	rate	integer	Рейтинг
	district	varchar	Район
	Ave_price	integer	Середній чек

Продовження таблиці 2.1

Назва таблиці	Назва стовпця	Тип даних	Опис даних
	imageURL	varchar	Посилання на зображення ресторану
Cuisine	cuisineID	integer	Код кухні, який ідентифікує її в БД
	name	varchar	Назва кухні
	imageURL	varchar	Посилання на зображення кухні
Dish	id	integer	Код страви, який ідентифікує її в БД
	name	varchar	Назва страви
	description	varchar	Опис страви
	rate	integer	Рейтинг страви
	price	integer	Ціна страви
	cuisineID	integer	Код кухні до якої відноситься страва
	restaurantID	integer	Код ресторану, якому належить страва
	imageURL	varchar	Посилання на зображення страви
Recommendation	id	integer	Код рекомендації, який ідентифікує її в БД
	restaurantID	integer	Код ресторану, якому належить страва
	dishID	integer	Код страви

Продовження таблиці 2.1

Назва таблиці	Назва стовпця	Тип даних	Опис даних
Preference	id	integer	Код уподобання, який ідентифікує його в БД
	dishID	integer	Код страви
	cuisineID	integer	Код кухні
	restaurantID	integer	Код ресторану
Review	id	integer	Код відгуку, який ідентифікує його в БД
	user_name	varchar	Ім'я користувача
	comment	varchar	Коментар користувача
	rate	integer	Оцінка користувача
	dishID	integer	Код страви
	restaurantID	integer	Код ресторану
	cuisineID	integer	Код кухні

**Висновок до розділу**

У даному розділі було розглянуто вхідні та вихідні дані для інформаційної системи пошуку ресторанів.

Вхідні дані надходять до системи з декількох джерел, а саме від: користувачів та адміністраторів. Вихідними даними є сформовані програмно рейтинги ресторанів та списки рекомендацій для користувача.

Також було наведено приклад вихідних даних у вигляді знімку екрану.

У розділі було описано структуру бази даних системи, для опису було використано ER-діаграму. На ER-діаграмі відображена функціональна залежність атрибутів сутностей та зв'язки між сутностями.

Також було сформовано таблицю, яка відображає структуру бази даних, містить інформацію про таблиці, які існують в базі даних, стовпці, типи даних та пояснення того, що міститься в базі даних.

Структура масивів інформації відповідає тим задачам, які повинна вирішувати система.

						Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

#### 3.1 Змістовна постановка задачі

Робота інформаційної система пошуку ресторанів зводиться до наступних задач : забезпечення релевантного пошуку по назві страви, кухні або ресторану в базі даних, забезпечення формування рейтингу найкращих ресторанів міста по обраному району та забезпечення користувача релевантними персональними рекомендаціями щодо вибору ресторану. Вищеперераховані функції є досить тривіальними і зводяться до нескладних операцій з базою даних, а саме: сортування, фільтрації та формування груп об'єктів.

Найскладнішою задачею є формування персональних рекомендацій для користувача. Тобто нам необхідно спрогнозувати та сформувати ряд пропозицій, найбільш близький до уподобань користувача та його попереднього досвіду. Також ми маємо врахувати досвід інших користувачів системи задля видачі більш цільових рекомендацій. Щоб реалізувати формування персональних рекомендацій застосуємо метод колаборативної фільтрації — доволі популярний метод, що застосовується у подібних рекомендаційних системах. Основою методу колаборативної фільтрації є твердження, що люди, які в минулому віддавали перевагу однаковим ресторанам схильні в майбутньому робити схожий вибір щодо того, який ресторан відвідати. Тобто проаналізувавши список уподобань та історію відвідувань і виставленої оцінки рейтингу ресторанів для кожного користувача системи ми зможемо формувати більш цільові рекомендації.

						Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2 Математична постановка задачі

Основною та цільовою задачею є формування релевантних рекомендацій користувачеві. Робота розроблюваної платформи пов'язана з такими сутностями як: ресторан, страва, кухня та користувач. Користувач отримує пропозицію у вигляді публікації страви з вказаною інформацією про саму страву, кухню та про ресторан.

Отже спочатку сформуємо постановку задачі.

Дано:

- $u$  — множина публікацій;
  - $n$  — кількість публікацій;
  - $m$  — кількість страв;
  - $i$  — конкретна публікація, де  $i = (1, n)$ ;
  - $j$  — конкретна страва, де  $j = (1, m)$ ;
  - $R_{i,j}$  — передбачувана оцінка публікації;
  - $R'_{i,j}$  — реальна оцінка публікації;
  - $u'$  — вектор публікації зі стравою з якою буде виконуватись порівняння;
- $\text{sim}(u, u')$  — функція, що визначає ступінь подібності публікацій;
- Цільова функція:

$$Z = \frac{1}{m} \sum_{j=1; m} (1 - \frac{|R_{i,j} - R'_{i,j}|}{\max(R_{i,j}, R'_{i,j})}) \rightarrow \max$$

де

$$R_{i,j} = \sum_{i=1; n} (\text{sim}(u, u') * R'_{i,j})$$

### 3.3 Обґрунтування методу розв'язання

Варто зазначити, що головна ціль інформаційної системи пошуку ресторанів — допомогти користувачеві знайти потрібний ресторан та сформувати для нього список рекомендацій.

У ситуації коли на нашу платформу заходить гість або ж користувач щойно зареєструвався і не додав жодної страви чи кухні у свій список уподобань, то він обов'язково отримає список рекомендацій, що сформований з найбільш популярних закладів міста.

У випадку коли користувач додає до списку своїх уподобань страву, яку до цього ще ніхто на платформі не додавав у список своїх уподобань, то система надасть користувачеві список рекомендацій сформований за Item-based методом колаборативної фільтрації. Тобто система проаналізує та знайде найбільш схожі страви, до тих, які є у списку уподобань користувача. А якщо в системі вже існують користувачі з подібними уподобаннями, то ми застосуємо гібридний метод колаборативної фільтрації. Гібридний метод поєднує у собі властивості Item-based та User-based методів.

Якщо користувач залишає відгук та оцінку страві, то система врахує виставлену оцінку користувачем та буде використовувати при формуванні рекомендацій. Тобто з'являється пріоритетність в стравах, чим вища оцінка, тим вищий пріоритет. У даному випадку буде застосований модифікований гібридний метод колаборативної фільтрації.

### 3.4 Опис методів розв'язання

Опишемо алгоритм застосування методу колаборативної фільтрації у вигляді 11 кроків.

Крок 1. Визначити вагові коефіцієнти для кожної існуючої публікації, врахувавши міру схожості

Крок 2. Вибрати зі списку публікацій ті, які є найбільш наближеними до списку вподобань

						Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Крок 3. Визначити прогнозований рейтинг на основі оцінок списку публікацій, який ми визначили на попередньому кроці

Крок 4. Визначити ваги кожної з публікацій на основі подібності властивостей враховуючи міру схожості

Крок 5. Вибрати список публікацій які за мірою схожості мають найбільші вагові коефіцієнти

Крок 6. Визначити прогнозовану оцінку схожості публікації взявши за основу схожі публікації

Крок 7. Визначити вагові коефіцієнти кожного з користувачів за схожістю їх уподобань , використавши міру схожості

Крок 8. Вибрати користувачів, які за мірою схожості мають найвищі показники

Крок 9. Визначити прогнозований рейтинг публікацій на основі уподобань користувачів з вибраного списку

Крок 10. Визначити вагові коефіцієнти для кожного з методів колаборативної фільтрації

Крок 11. Визначити фінальний результуючий рейтинг публікації

### Висновок до розділу

У розділі математичного забезпечення було визначено змістовне і математичне формулювання завдання формування рекомендацій на основі методу колаборативної фільтрації. Розглянуті кілька підходів до вирішення поставленої задачі.

Описано та обгрунтовано вибір даного методу.

Детально описаний алгоритм визначення релевантних рекомендацій.

						Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19



## 4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

### 4.1 Засоби розробки

Для створення програмного продукту були застосовані такі засоби програмування, як:

- HTML, CSS, JavaScript, Bootstrap — для розробки клієнтської частини та Docker для розгортання клієнтської частини;
- TypeScript, Node.js, EJS, Express.js для серверної частини, а також була використана база даних MongoDB та система управління Redis.
- Для коректної взаємодії сайту та веб-платформи, було реалізовано API, що приймає запити з сайту та передає дані до сервісу.

Середовище розробки — Visual Studio Code.

JavaScript (JS) — динамічна, об'єктно-орієнтована, прототипна мова програмування загального призначення. JavaScript є реалізацією стандарту ECMAScript. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінок [19].

TypeScript — мова програмування, що позиціонує себе як засіб для розробки веб-застосунків, що суттєво розширює можливості JavaScript. Тобто, TypeScript є сумісною з JavaScript, головною особливістю TypeScript є чітка типізація, також використовуючи дану мову помітно збільшується швидкість розробки через повторне використання коду та рефакторингу коду. Фактично, після компіляції програму на TypeScript можна виконувати в будь-якому сучасному браузері або використовувати спільно з серверною платформою Node.js [9].

Node.js — програмна платформа з відкритим кодом, що використовується для виконання високопродуктивних мережевих веб-

						Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

застосунків, що написані на мові JavaScript. Node.js надає можливість виконувати скрипти JavaScript на сервері та відправляти результат їх виконання [3].

Express.js (Express) — фреймворк веб-застосунків для Node.js, реалізований як вільне і відкрите програмне забезпечення під ліцензією MIT. Він спроектований для створення веб-застосунків і API. Express фактично є стандартним програмним каркасом для Node.js [10].

EJS — шаблонізатор для платформи Node.js. Його перевагами є простота у використанні і проста інтеграція з модулем Express для Node.js. Недоліки присутні в його малій функціональності і відсутності офіційного постачальника. Проте функціональність компенсується за рахунок функціоналу з Express.js [11].

MongoDB — документо-орієнтована система управління базами даних класу NoSQL з відкритим вихідним кодом, яка не вимагає опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними в форматі ключ / значення, і реляційними СУБД, функціональними і зручними у формуванні запитів. Також у парі з MongoDB працює Redis [12].

Redis (Remote Dictionary Server) — розподілене сховище пар ключ-значення, які зберігаються в оперативній пам'яті, з можливістю забезпечувати довговічність зберігання за бажанням користувача [13].

Docker — програмне забезпечення для автоматизації розгортання і управління додатками в середовищах з підтримкою контейнеризації [20]. Docker дозволяє «упакувати» додаток з усім його оточенням і залежностями в контейнер даних [14].

## 4.2 Вимоги до технічного забезпечення

### 4.2.1 Загальні вимоги

Для коректної роботи розробленої програми до складу технічних засобів повинні входити:

– комп'ютер з такою конфігурацією:

- 1) процесор з тактовою частотою не нижче 1 ГГц;
- 2) об'єм RAM (не менше 256 МБ);
- 3) до інших складових вимоги не висуюються, бо вони несуттєво впливають на роботу програми;

– додатково має бути встановлене таке програмне забезпечення:

- 1) операційна система Windows/Linux/OSX;
- 2) середовище розробки Visual Studio Code;
- 3) браузер Google Chrome/ Mozilla Firefox/ Opera;

– комп'ютерна периферія, до складу якої входять:

- 1) монітор;
- 2) мишка;
- 3) клавіатура.

## 4.3 Архітектура програмного забезпечення

Архітектура програми умовно поділяється на дві частини: front–end частину та back–end частину.

Front–end частина розроблена на базі фреймворку Redis. Front–end відповідає за отримання та опрацювання даних, що надходять з сервера.

Back–end частина побудована за допомогою технологій Node.js та фреймворку Express. Ця частина відповідає за відправлення запрошених даних та роботу з базою даних MongoDB.

На рисунку 4.1 зображено структуру програмного забезпечення.

						Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

### Інтерактивна система

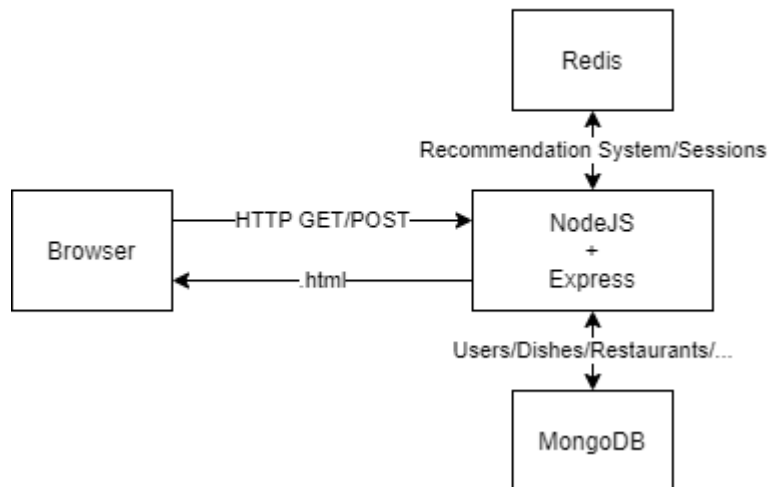


Рисунок 4.1. Структура програмного забезпечення

#### 4.3.1 Діаграма класів

Схема структурна класів представлена у частині графічного матеріалу.

Діаграма (серверна частина) містить 5 класів, а саме:

- “User” — клас програми, що визначає користувачів системи та містить дані про них: ім’я, e-mail, пароль та роль (користувач/адміністратор). Користувачі мають можливість реєструватися, входити/ виходити з системи, вносити страви та кухні в список уподобань, залишати відгуки та оцінки стравам і ресторанам;
- “Review” — клас програми, що визначає відгуки користувачів та містить дані про користувача, страву, оцінку (виставлений рейтинг) та текст відгуку;
- “Restaurant” — клас програми, що визначає ресторан та містить дані про назву ресторану, адресу, контактний номер, веб-сайт, район та посилання на зображення ресторану;

- “Kitchen” — клас програми, що визначає кухню та містить дані про назву кухні та посилання на зображення кухні;
- “Dish” — клас програми, що визначає страву та містить дані про назву страви, опис, ціну, кухню, ресторан, у якому готують страву та посилання на зображення страви.

#### 4.3.2 Діаграма послідовності

Схема структурна послідовності наведена у розділі графічного матеріалу. На діаграмі відображена послідовність дій, виконуваних користувачем програми та адміністратором.

#### 4.3.3 Діаграма компонентів

Схема структурна компонентів зображена у розділі графічного матеріалу. На даній діаграмі відображені компоненти, що використовуються при розробці архітектури, та взаємозв’язки між ними. Основними компонентами в системі є користувач, веб-сайт, веб-інтерфейс та сервер. Користувачі залишають запит на отримання рекомендацій на веб-сайті, далі запит відправляється на веб-інтерфейс, який у свою чергу передає дані на обробку сервера.

#### 4.3.4 Специфікація функцій

Функції класів програмного забезпечення наведені в таблиці 4.1.

Таблиця 4.1 – Функції класів програмного забезпечення

Назва	Примітка
Клас Dish — визначає страву	
getTopDishes()	Виконує відображення найбільш популярних страв
findDishesByName()	Реалізує пошук страви за назвою
findDishesByKitchen()	Реалізує пошук страви за кухнею
addDish()	Виконує додавання страви

					Арк.
Змн.	Арк.	№ докум.	Підпис	Дата	24

Продовження таблиці 4.1

Назва	Примітка
deleteDish()	Виконує видалення страви
getCollaborativeRecommendations()	Реалізує підбір рекомендацій зі стравою
Клас Kitchen — визначає кухню	
getKitchens()	Виконує відображення усіх кухонь
addKitchen()	Виконує додавання кухні
deleteKitchen()	Виконує видалення кухні
Клас Restaurant — визначає ресторан	
getRestaurants()	Виконує відображення усіх ресторанів
getRestaurantsMenu()	Виконує відображення меню ресторану
addRestaurant()	Виконує додавання ресторану
deleteRestaurant()	Виконує видалення ресторану
getRestaurantsSortedByRating()	Виконує відображення найбільш популярних ресторанів, відсортованих за рейтингом
Клас Review — визначає відгук	
findReviewsByRestaurant()	Виконує відображення відгуків про ресторан
addReview()	Виконує додавання відгуку
Клас User — визначає користувача(адміністратора)	
getUsers()	Виконує відображення списку всіх користувачів
findUserByEmail()	Реалізує пошук користувача за e-mail
registerUser()	Реалізує реєстрацію користувача
deleteUser()	Виконує видалення користувача
setAdminUser()	Виконує призначення користувача адміністратором

## Висновок до розділу

У цьому розділі було визначено програмне забезпечення для розробки Інформаційної системи пошуку ресторанів. Для розробки програмного продукту було використано наступні засоби програмного забезпечення: TypeScript, Node.js, EJS, Express.js, типовий набір для розробки клієнтської частини — HTML, CSS, JS, Bootstrap, для роботи з даними була обрана NoSQL база даних MongoDB у парі з системою управління базою Redis. Для розгортання продукту був використаний Docker. У розділі був обґрунтований вибір вищеперерахованих програмних засобів та висвітлені основні переваги їх використання. Також було наведено загальні вимоги до технічного забезпечення.

Для представлення архітектури програмного забезпечення побудовано діаграми класів, послідовності та компонентів з детальним описом цих діаграм.

Також були наведені специфікації функцій, які задіяні в роботі програми задля формування рекомендацій користувачеві.

						Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

## 5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

### 5.1 Керівництво користувача

Для запуску програмного застосування необхідно в командному рядку відкрити кореневу папку проекту та прописати команду `docker-compose up`, а потім у пошуковому рядку браузера прописати : `localhost:3000`

Коли користувач вперше заходить на веб-платформу він потрапляє на головну сторінку. На головній сторінці міститься навігаційна панель, поле для введення пошукового запиту, форма реєстрації та авторизації (рисунок 5.1).

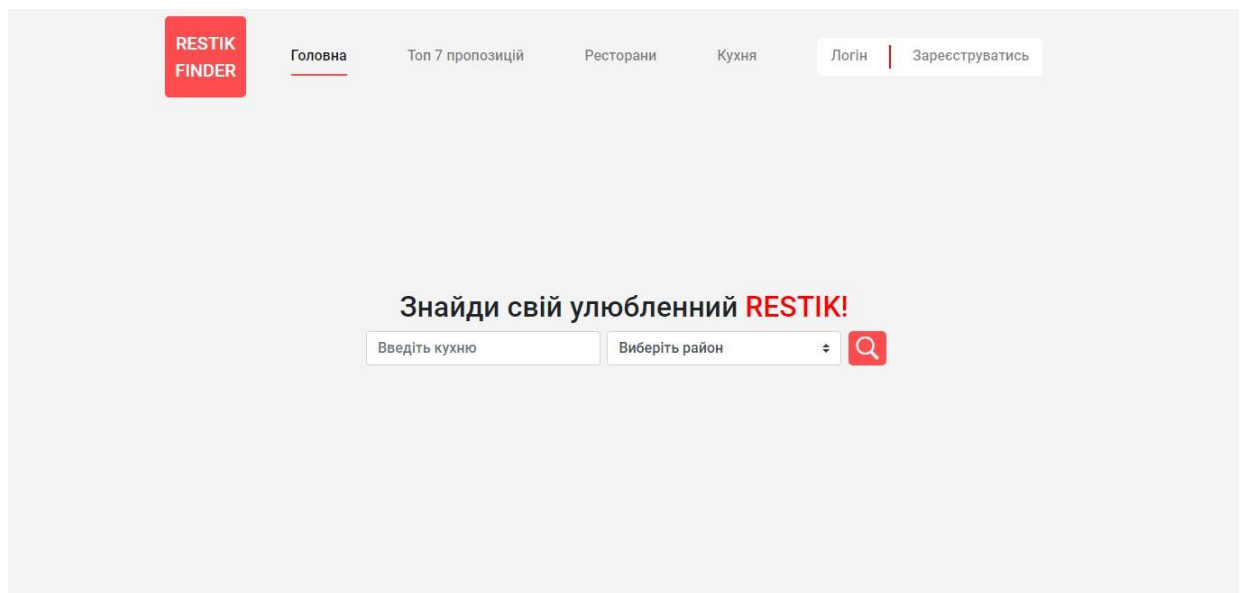


Рисунок 5.1. Головна сторінка

Система передбачає користування для кількох типів користувачів: гостя, зареєстрованого користувача та адміністратора.

Незареєстрований користувач, тобто гість платформи має можливість виконувати пошук ресторану, може переглядати найкращі пропозиції, ресторани та кухні, які опубліковані в системі.

Гість платформи має можливість авторизуватись або зареєструватись у системі натиснувши на відповідні пункти навігаційної панелі «Логін» або «Зареєструватись». Після натискання на кнопки «Логін» чи

						Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		



«Зареєструватись» – на сторінці відобразиться поп-ап вікно для введення необхідних даних. (рисунок 5.2-5.3).

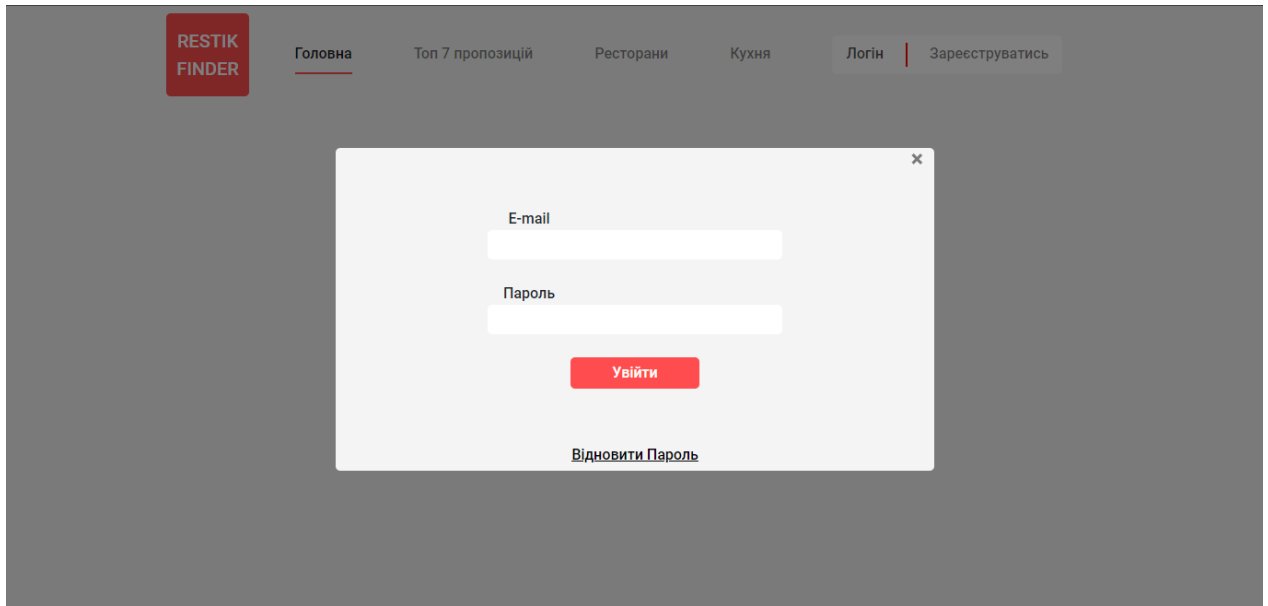


Рисунок 5.2. Сторінка авторизації

У поп-ап вікні авторизації користувач має ввести особисті дані, які уже збережені в базі даних (E-mail та Пароль) для того, щоб увійти в систему. Для коректної авторизації у системі, виконується перевірка на

- заповненість полів;
- наявність введених e-mail та паролю в базі даних системи.

У разі успішної авторизації користувач потрапляє на головну сторінку з розширеною навігаційною панеллю та розширеними можливостями користування. (рисунок 5.5), інакше, користувач отримає повідомлення, що його обліковий запис не знайдено у базі даних, неправильно введено e-mail та/або пароль. Користувач має можливість повторно ввести необхідні дані для спроби входу, кількість спроб не обмежена.

Приклад авторизації з помилкою зображено на рисунку 5.3.

						Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

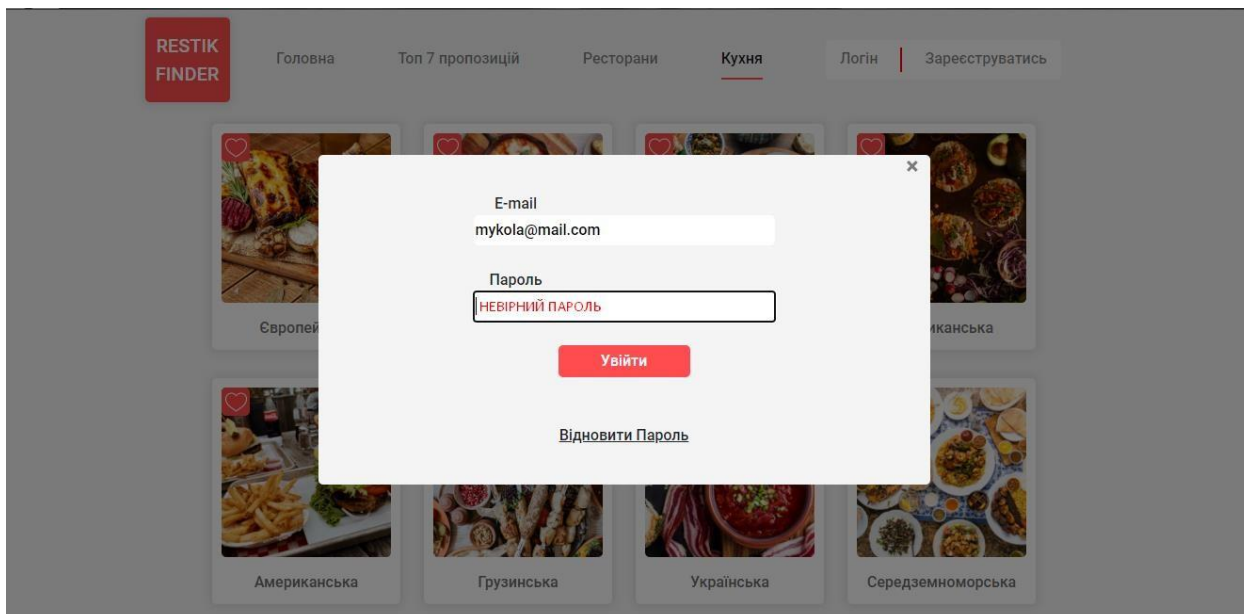


Рисунок 5.3. Помилка при авторизації

Також користувач має можливість відновити доступ до свого облікового запису, для цього він має натиснути на посилання «Відновити пароль» — пароль буде надіслано на вказаний e-mail користувача.

Гість системи має можливість зареєструватися у системі, для цього він має обрати пункт «Зареєструватись» на навігаційній панелі. Для успішної реєстрації користувачеві необхідно ввести своє ім'я, e-mail та пароль.

Приклад сторінки реєстрації зображений на рисунку 5.4

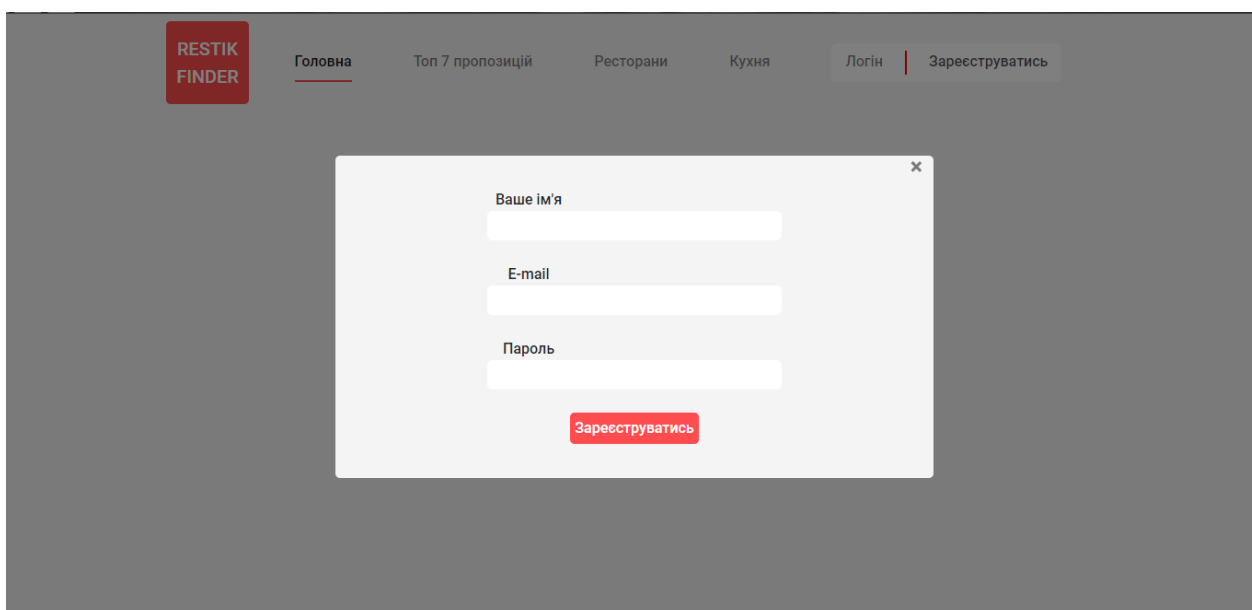


Рисунок 5.4. Сторінка реєстрації

						Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Для коректної реєстрації у системі, виконується перевірка на

- заповненість полів;
- коректність введення електронної пошти (e-mail має містити знак @, наприклад, [user007@restik-finder.com](mailto:user007@restik-finder.com));
- наявність зареєстрованого користувача з введеним e-mail.

У разі успішної авторизації користувач потрапляє на головну сторінку з розширеною навігаційною панеллю та розширеними можливостями користування. (рисунк 5.5), інакше, користувач отримає повідомлення про помилку реєстрації.

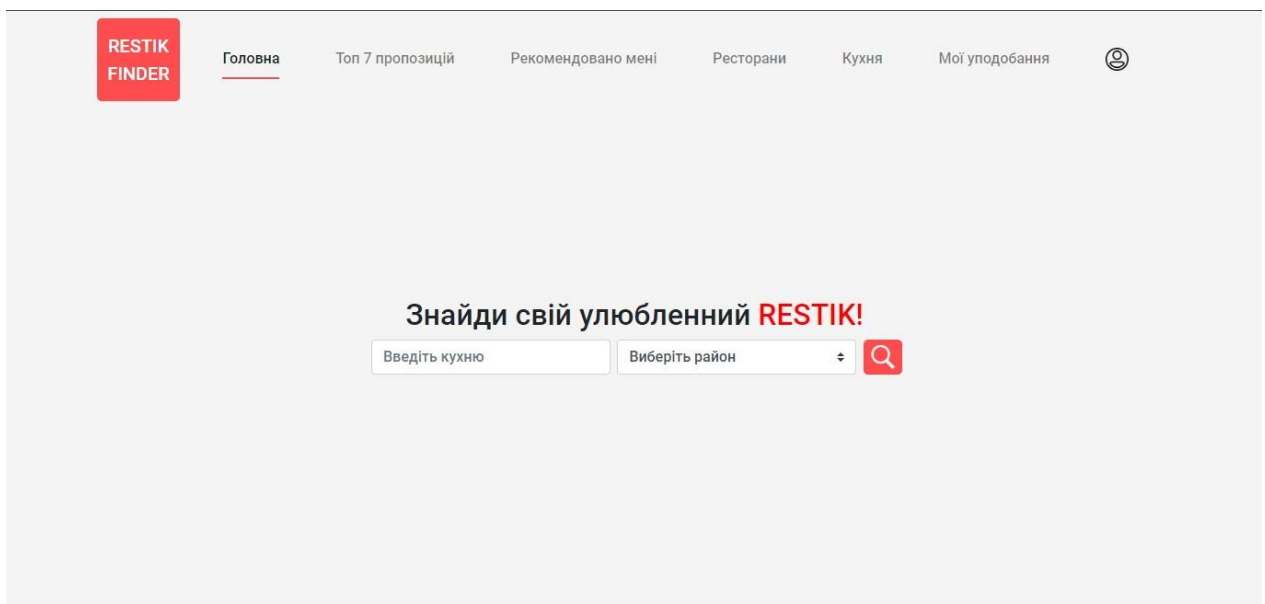


Рисунок 5.5. Головна сторінка зареєстрованого користувача

Перш за все гість та зареєстрований користувачі можуть виконати пошук ресторану, страви чи кухні, для цього необхідно ввести пошуковий запит у відповідне поле для пошуку та натиснути кнопку пошуку або клавішу Enter на клавіатурі. Приклад результату пошукового запиту «салат» зображено на рисунку 5.6

						Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

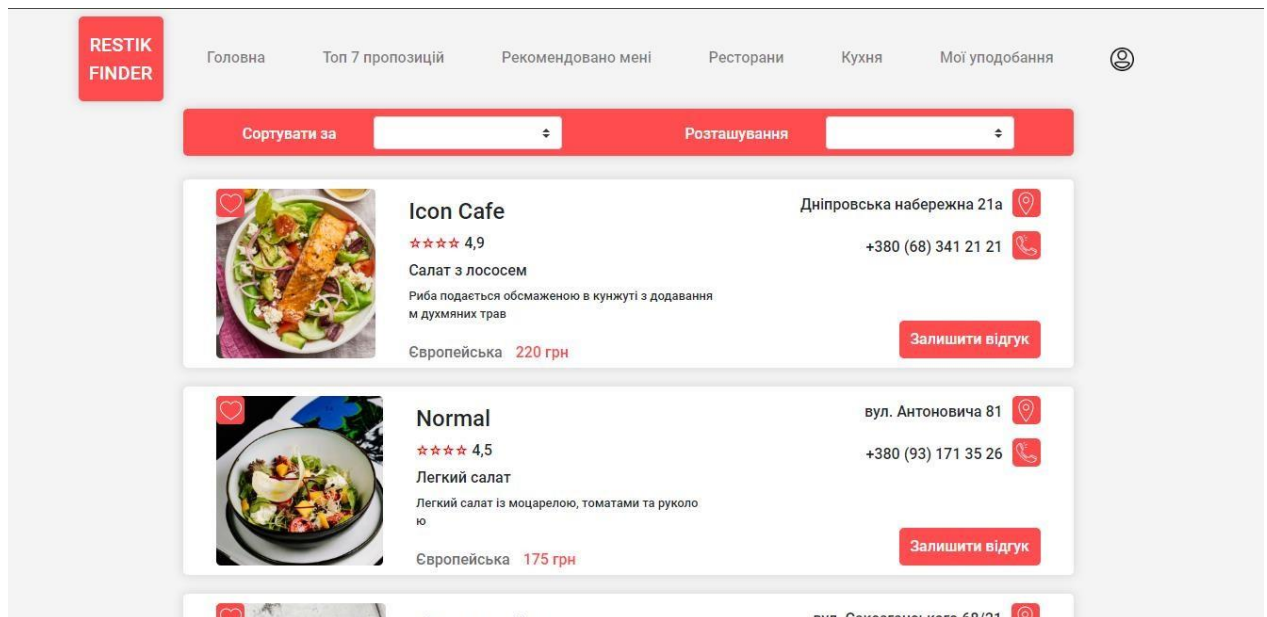


Рисунок 5.6. Результат виконаного пошуку

Авторизований користувач та гість мають можливість переглянути списки пропозицій від ресторанів, переглянути кухні та переглянути топ 7 пропозицій від платформи.

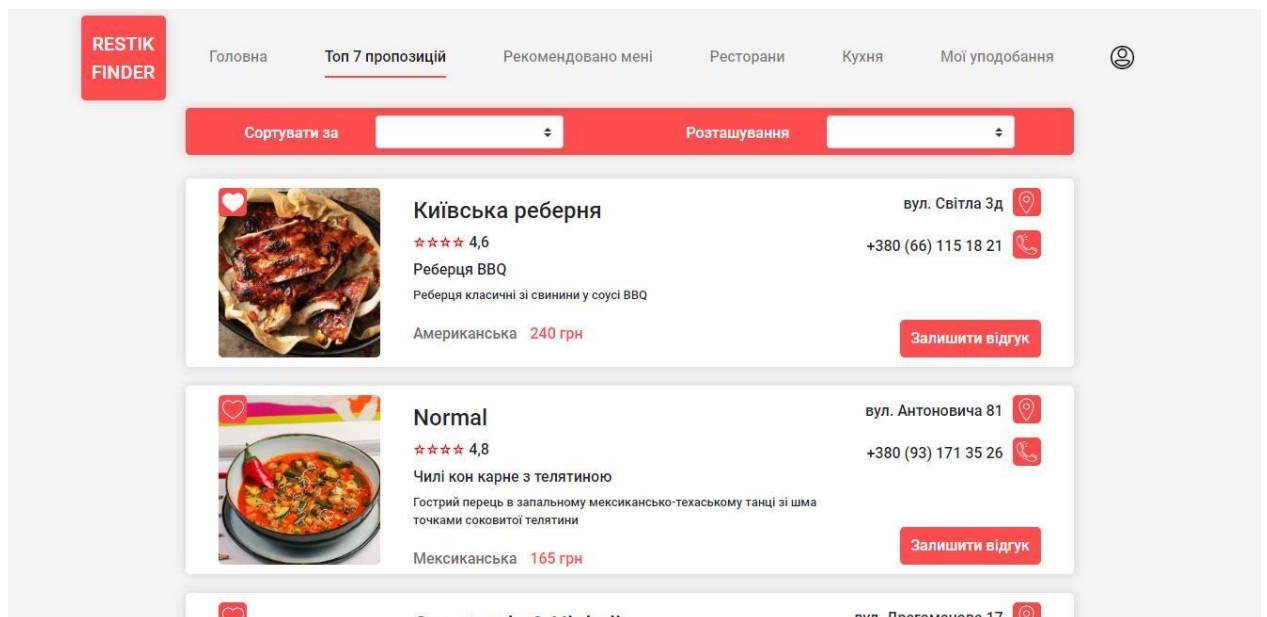


Рисунок 5.7. Сторінка з топ 7 пропозицій платформи

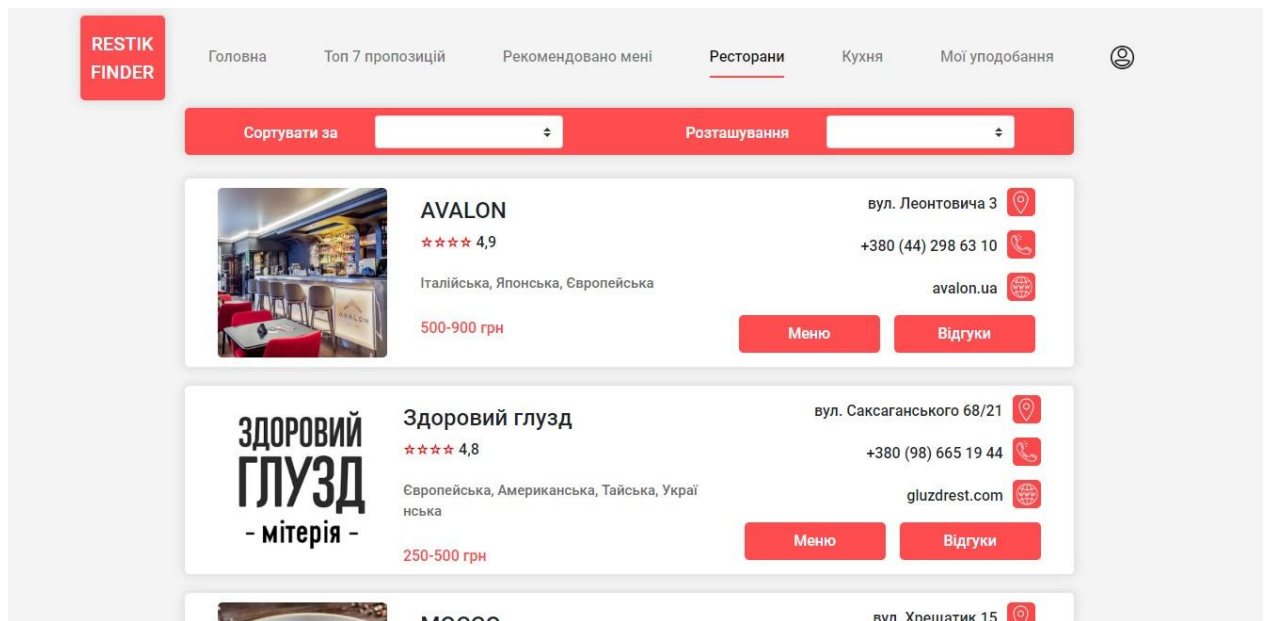


Рисунок 5.8. Сторінка з ресторанами міста

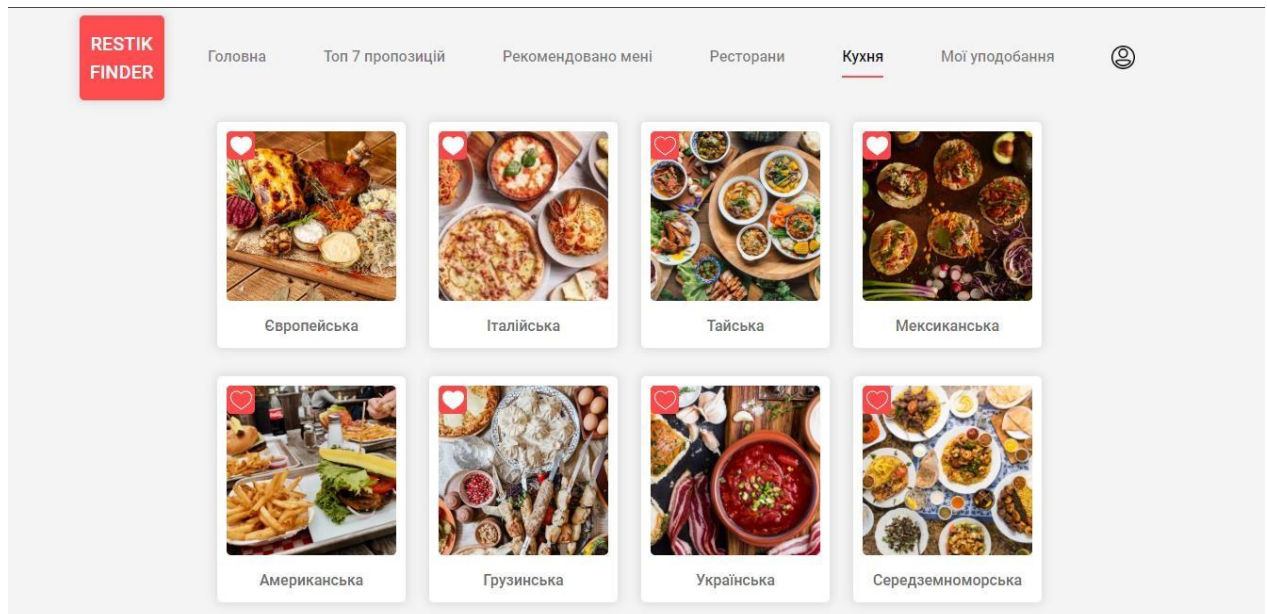


Рисунок 5.9. Сторінка з кухнями

Авторизований користувач має можливість додавати пропозиції ресторанів та кухні у список своїх уподобань, натиснувши на кнопку «♥». Уподобання користувача містяться в особистому кабінеті користувача та на сторінці «Мої уподобання»

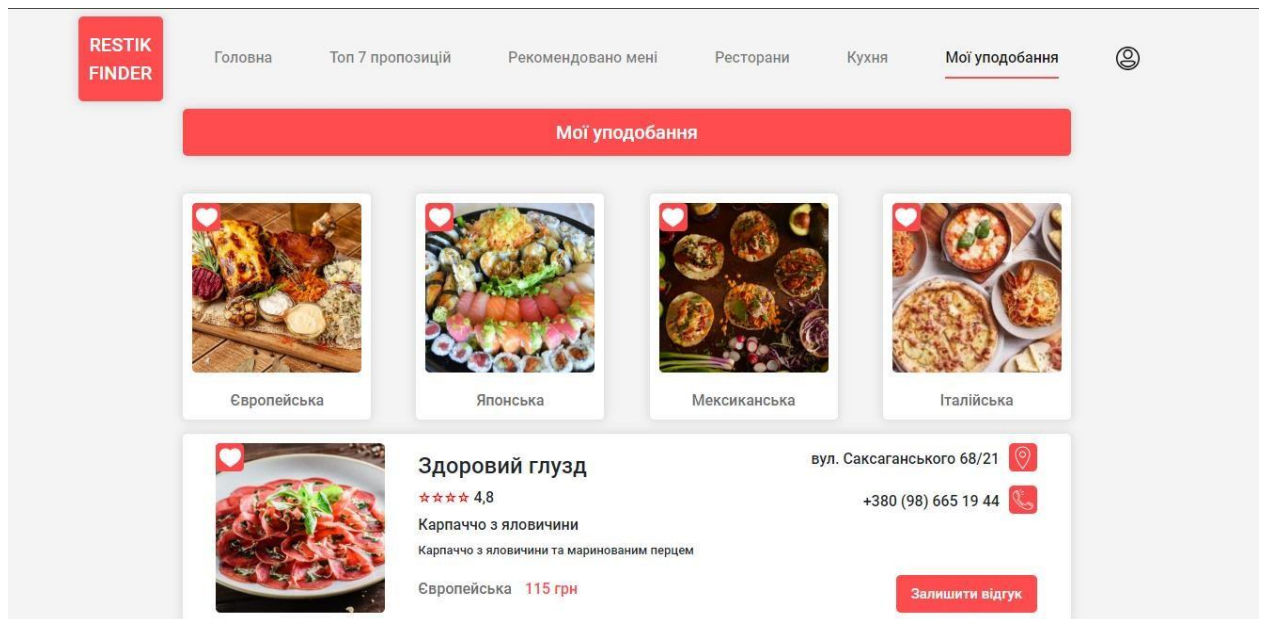


Рисунок 5.10. Сторінка з уподобаннями користувача

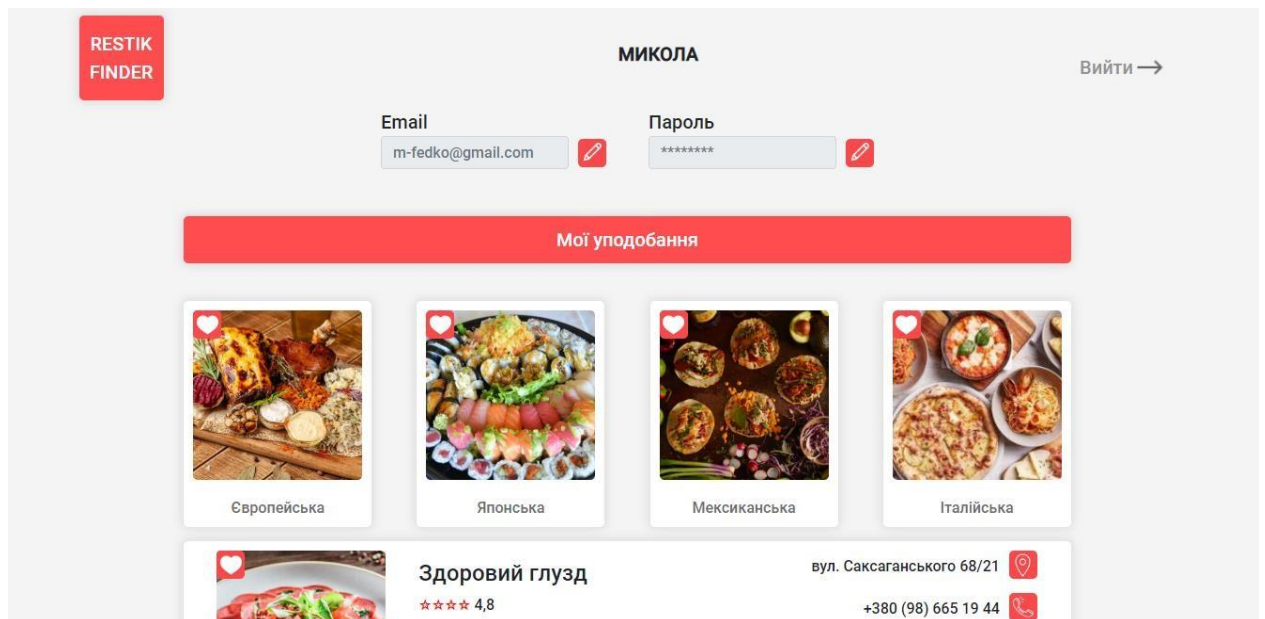


Рисунок 5.11. Особистий кабінет користувача

У особистому кабінеті користувач має можливість змінити власний пароль та e-mail, також користувач може видаляти страви та кухні зі списку своїх уподобань, натиснувши на кнопку «♥».



Користувач має можливість сортувати та фільтрувати публікації на сторінках з пропозиціями, ресторанами та персональними рекомендаціями. Приклад результату сортування зображено на рисунках 4.12 та 4.13

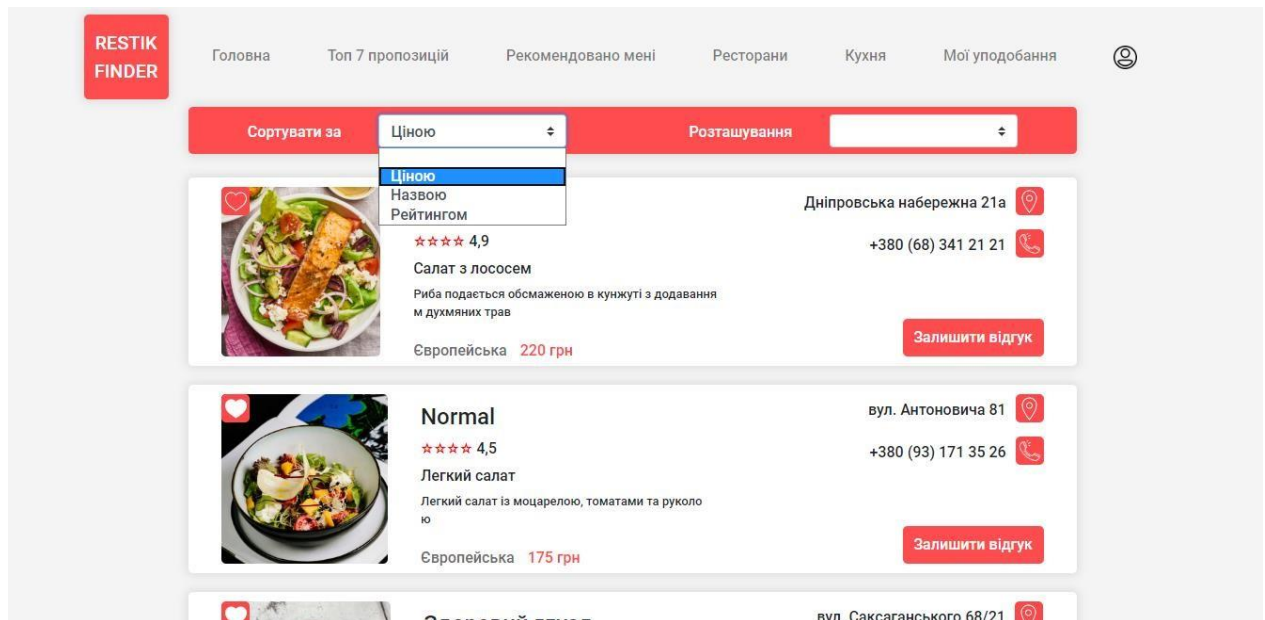


Рисунок 5.12. Вибір критерію сортування

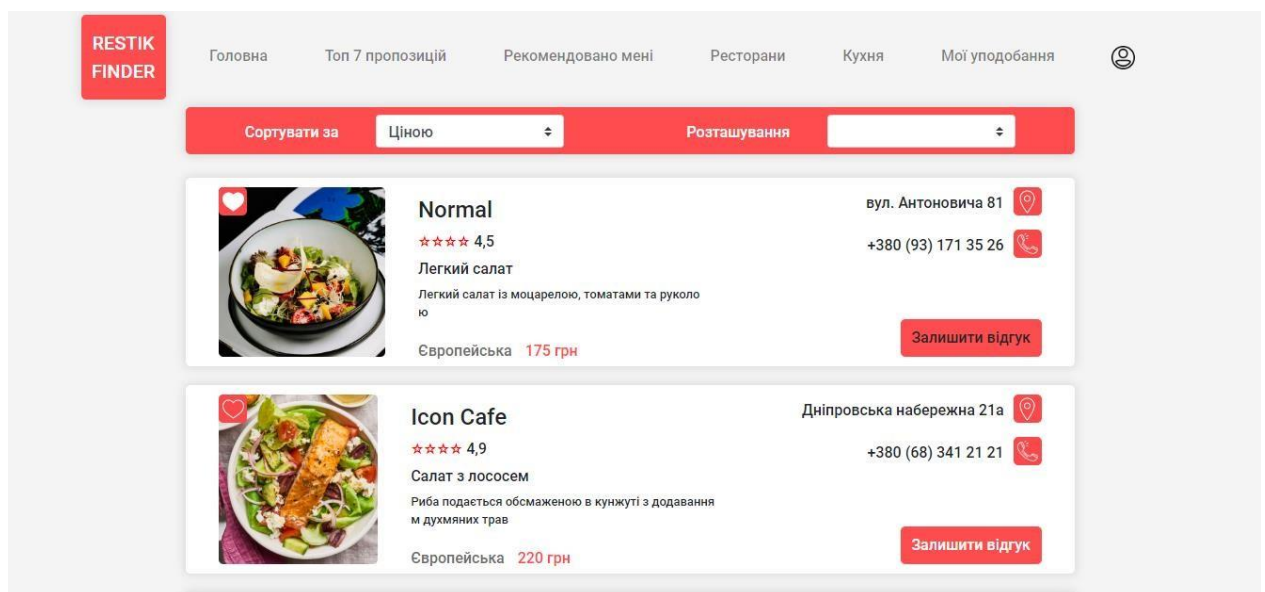


Рисунок 5.13. Пропозиції відсортовані за ціною

Користувач має можливість переглянути рекомендації від платформи, для цього необхідно обрати пункт «Рекомендовано мені» з навігаційної панелі. Приклад рекомендацій зображено на рисунку 5.14

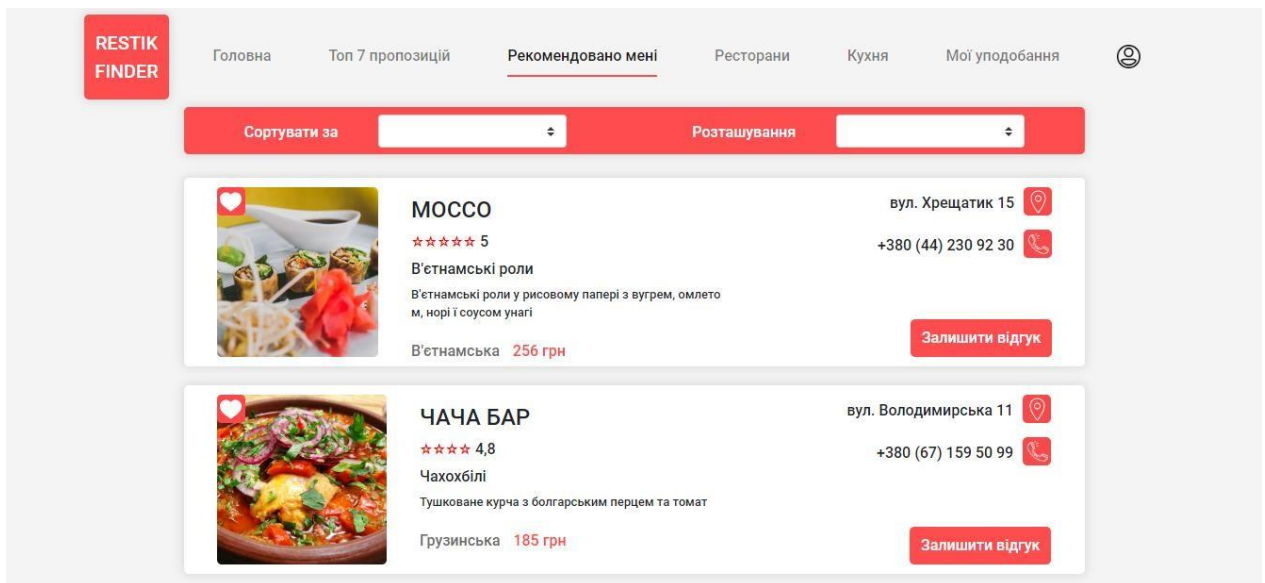


Рисунок 5.14. Список рекомендацій для користувача

Інформаційна система пошуку ресторанів також передбачає користування та управління платформою адміністратором. Адміністратор може зайти в свій особистий кабінет увівши необхідні дані для авторизації у вікні авторизації (рисунок 5.2). Після успішної авторизації адміністратор потрапляє в особистий кабінет адміністратора (рисунок 5.15).

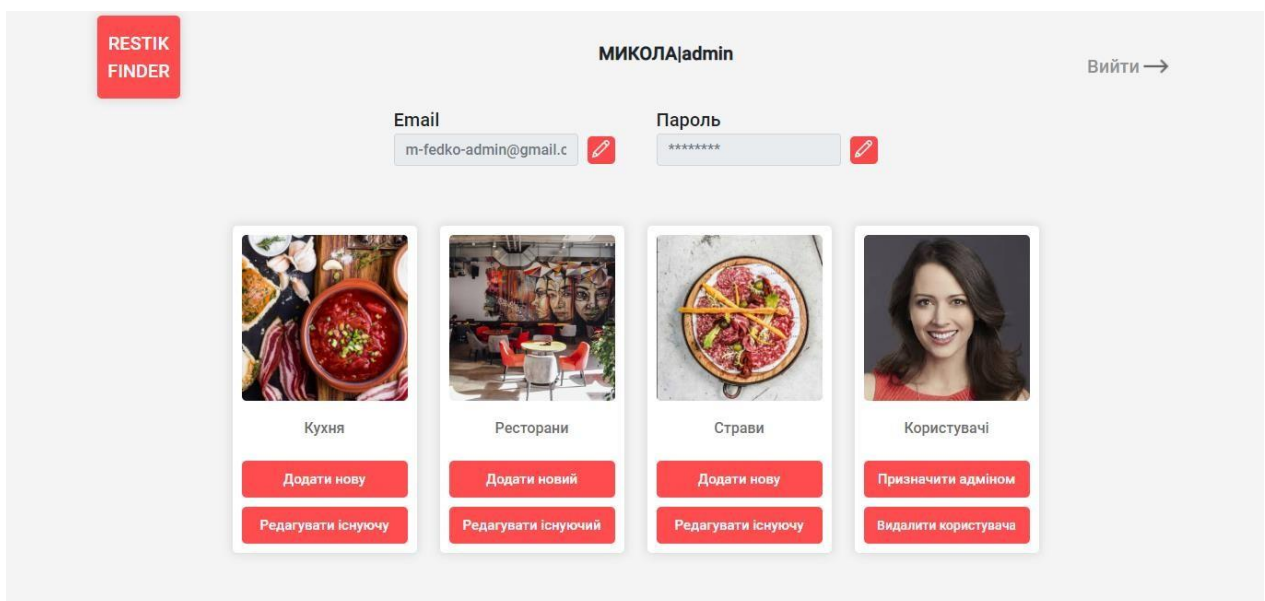


Рисунок 5.15. Особистий кабінет адміністратора



У особистому кабінеті адміністратор має можливість додавати нові, видаляти та редагувати існуючі публікації зі стравами, кухнями та ресторанами, а також може надавати доступ адміністратора іншим користувачам та видаляти облікові записи користувачів. Приклад форми додавання нового ресторану в систему зображено на рисунку 5.16

RESTIK FINDER

МИКОЛА|admin

Вийти →

Назва

Адреса

Район

Контактний номер

Веб-Сайт

Зображення ресторану

Рисунок 5.16. Форма додавання нового ресторану

Адміністратор має можливість редагувати інформацію про ресторани, кухні та страви, приклади редагування зображені на рисунках 5.17 – 5.20

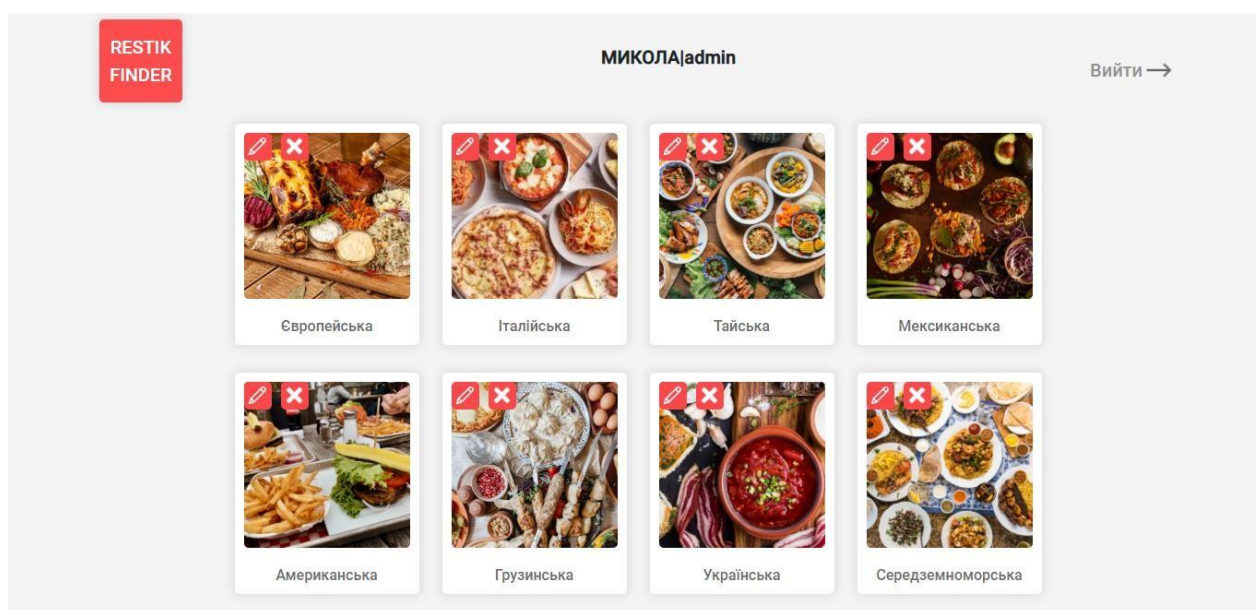


Рисунок 5.17. Сторінка вибору кухні для редагування

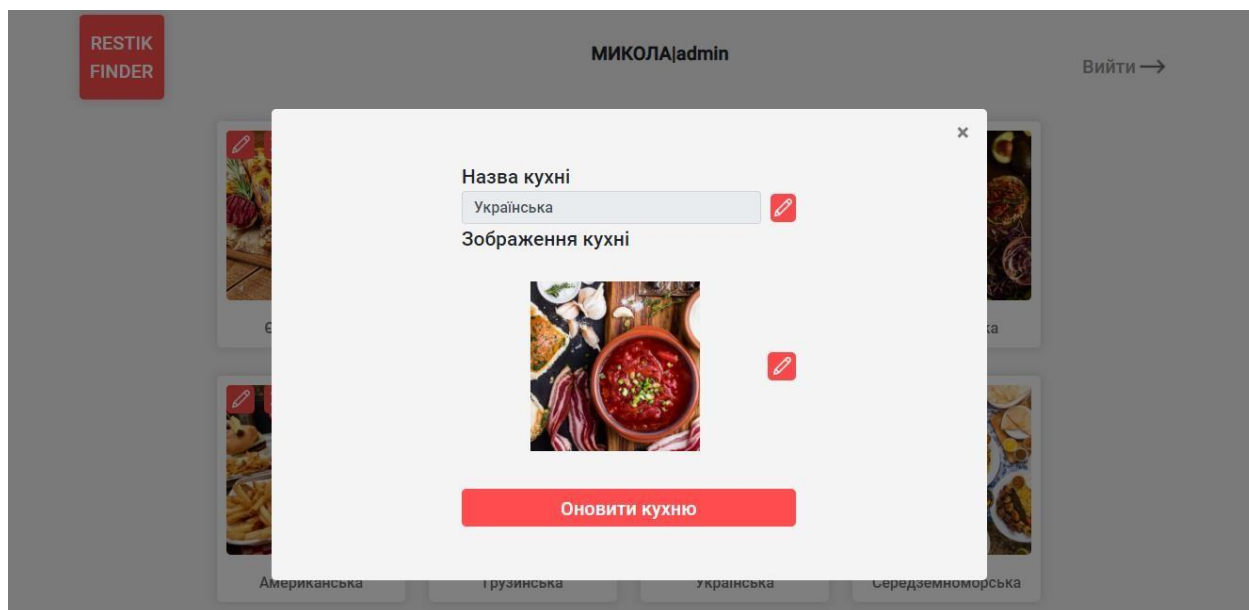


Рисунок 5.18. Форма редагування інформації про кухню

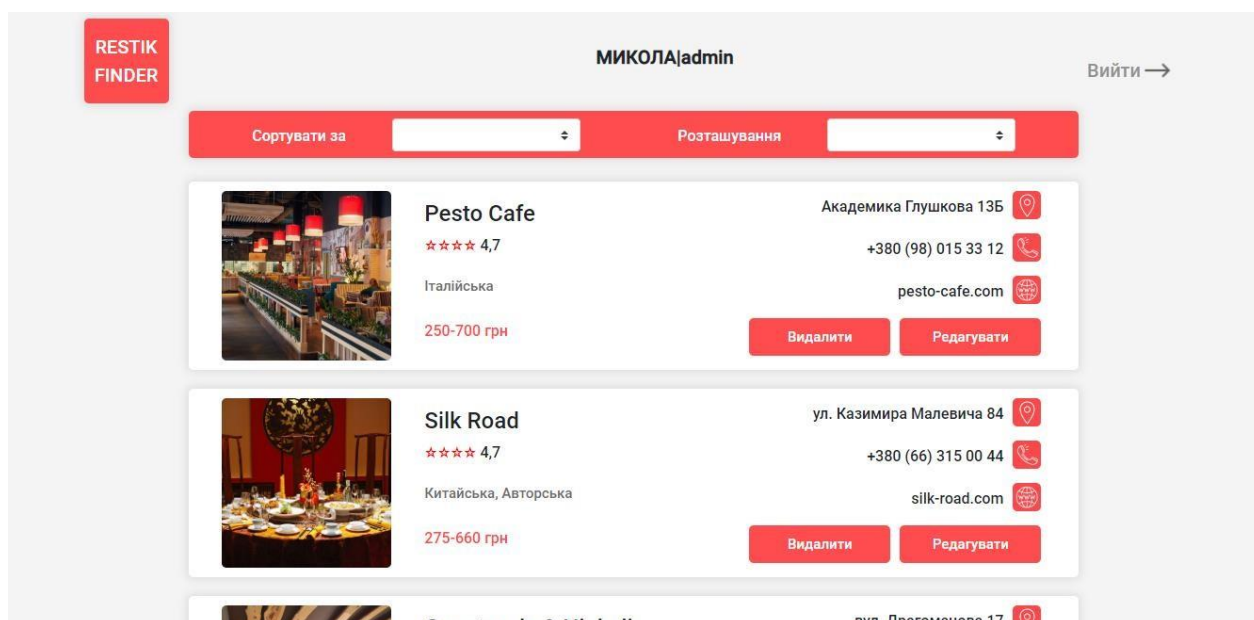


Рисунок 5.19. Сторінка вибору ресторану для редагування або видалення

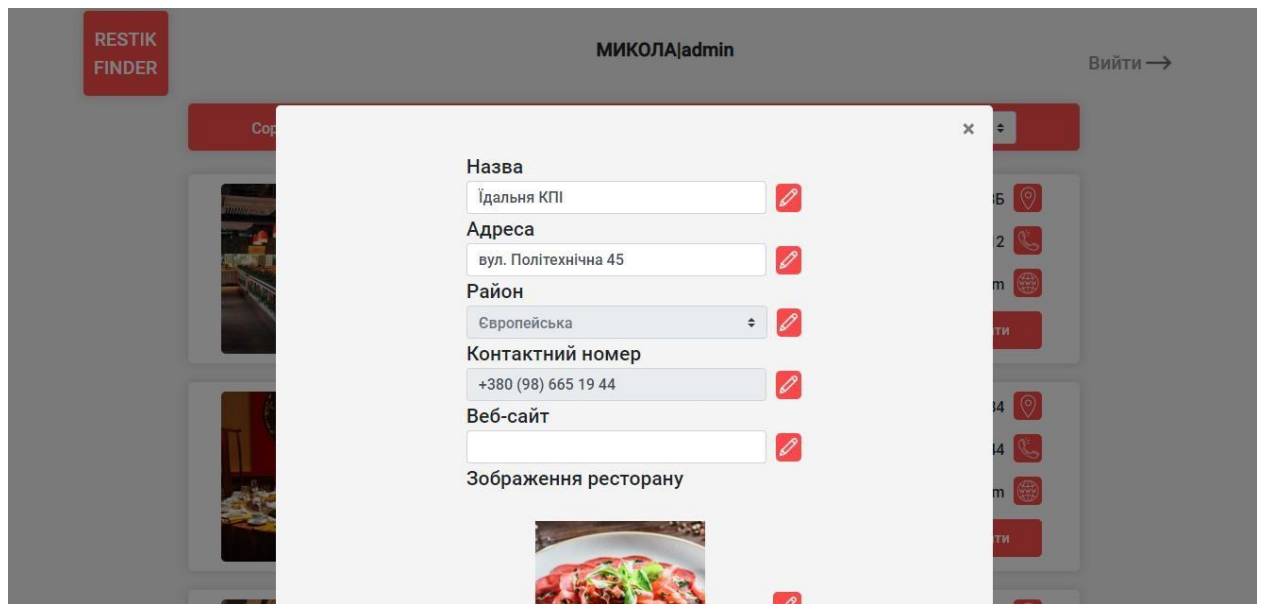


Рисунок 5.20. Форма редагування інформації про ресторан

Адміністратор має можливість надати права доступу адміністратора для іншого зареєстрованого користувача, тобто може призначити новим адміністратором платформи. Приклад сторінки призначення нового адміністратора зображено на рисунку 5.23

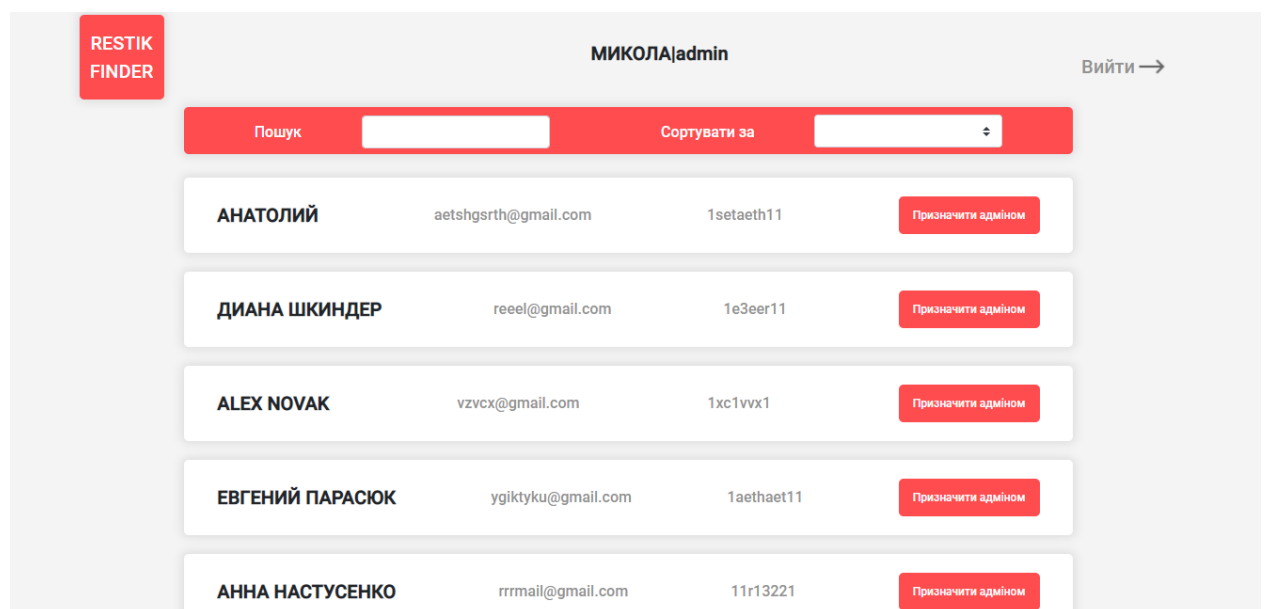


Рисунок 5.21. Сторінка призначення нового адміністратора

Адміністратор має можливість видалити користувача з бази даних інформаційної системи пошуку ресторанів. Приклад сторінки для видалення користувачів зображено на рисунку 5.24

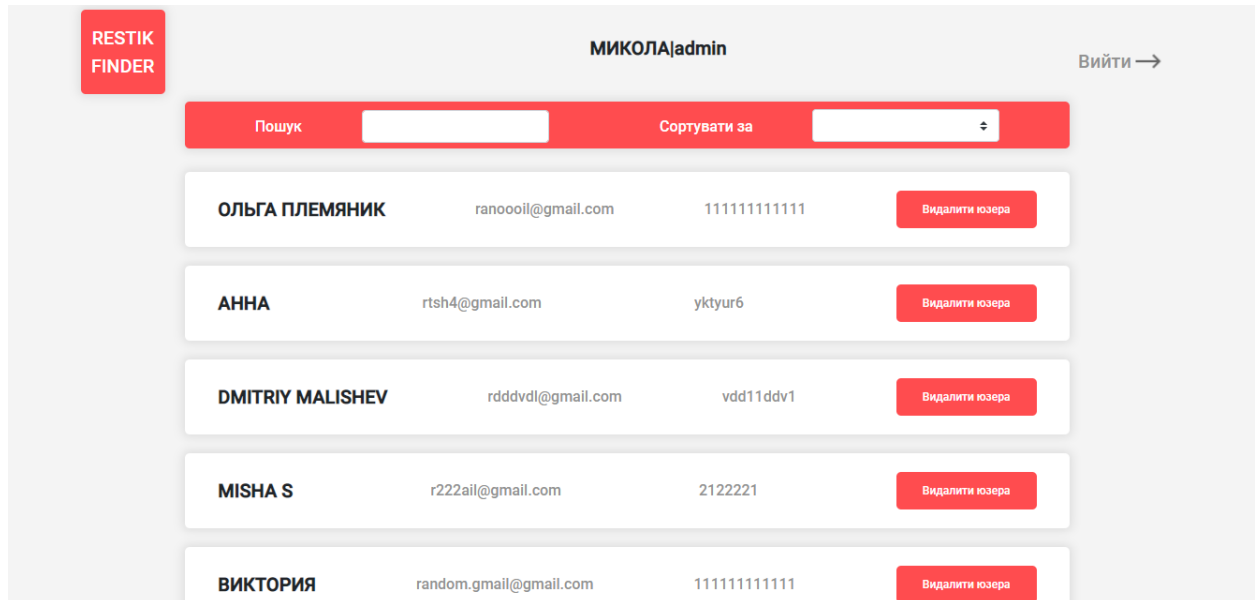


Рисунок 5.22. Сторінка видалення користувача

Користувач та адміністратор можуть вийти з особистого кабінету натиснувши на кнопку «Вийти». При виході з особистого кабінету всі введені та змінені дані зберігаються в системі.

## 5.2 Випробування програмного продукту

У цьому підрозділі наведено опис проведених тестів та порядок їх виконання. Тести проводяться задля перевірки відповідності розробленого програмного забезпечення функціональним вимогам, представлених у технічному завданні на створення інформаційної системи пошуку ресторанів.

### 5.2.1 Мета випробувань

Метою випробувань є перевірка відповідності функцій інформаційної системи пошуку ресторанів вимогам технічного завдання, перевірка

						Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

коректності роботи розробленого програмного продукту та його взаємодії з базою даних.

### 5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

### 5.2.3 Результати випробувань

Під час виконання тестування було перевірено всю функціональність системи. У наступних таблицях наведено список випробувань базових функціональних можливостей програмного продукту (табл. 5.1 – 5.34).

Таблиця 5.1 – Перевірка функції пошуку ресторану для неавторизованого користувача

Мета тесту	Перевірка функції пошуку ресторану
Початковий стан	Відкрита «Головна» сторінка сайту (без попередньої авторизації)
Вхідні дані	Пошуковий запит користувача
Схема проведення тесту	Користувач вводить пошуковий запит в відповідне поле та/або обирає бажаний район з випадаючого списку для фільтрації результатів
Очікуваний результат	Відображення списку релевантних публікацій з пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка зі списком релевантних публікацій з пропозиціями від ресторанів

Таблиця 5.2 – Перевірка функції сортування результатів пошуку

Мета тесту	Перевірка функції сортування результатів пошуку
Початковий стан	Відкрита сторінка з результатами пошуку
Схема проведення тесту	Користувач обирає бажаний критерій сортування списку публікацій із випадального списку: «за рейтингом», «за назвою» чи «за ціною»
Очікуваний результат	Відображення відсортованого списку публікацій з пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка з відсортованим списком публікацій з пропозиціями від ресторанів

Таблиця 5.3 – Перевірка функції фільтрації результатів пошуку за районом

Мета тесту	Перевірка функції фільтрації результатів пошуку
Початковий стан	Відкрита сторінка з результатами пошуку
Схема проведення тесту	Користувач обирає бажаний район для фільтрації списку публікацій
Очікуваний результат	Відображення фільтрованого за обраним районом списку публікацій із пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка з фільтрованим за обраним районом списком публікацій із пропозиціями від ресторанів

Таблиця 5.4 – Перевірка відображення сторінки «Топ 7 пропозицій» для неавторизованого користувача

Мета тесту	Перевірка відображення сторінки «Топ 7 пропозицій»
Початковий стан	Відкрита «Головна» сторінка сайту
Схема проведення тесту	Користувач натискає на пункт меню «Топ 7 пропозицій»
Очікуваний результат	Відображення сторінки з 7 найпопулярнішими пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка з 7 найпопулярнішими пропозиціями від ресторанів

Таблиця 5.5 – Перевірка функції «Залишити відгук» для неавторизованого користувача

Мета тесту	Перевірка функції «Залишити відгук» для неавторизованого користувача
Початковий стан	Відкрита сторінка сайту «Топ 7 пропозицій»
Схема проведення тесту	Користувач натискає кнопку «Залишити відгук» до бажаної публікації
Очікуваний результат	Відображення попередження, що функція доступна для авторизованих користувачів та прохання авторизуватись або зареєструватись
Стан після проведення випробувань	Відображена сторінка сайту «Топ 7 пропозицій»

Таблиця 5.6 – Перевірка функції додавання страви до списку уподобань для неавторизованого користувача

Мета тесту	Перевірка функції додавання страви до списку уподобань для неавторизованого користувача
Початковий стан	Відкрита сторінка сайту «Топ 7 пропозицій»
Схема проведення тесту	Користувач натискає кнопку «♥» біля зображення бажаної страви
Очікуваний результат	Відображення попередження, що функція доступна для авторизованих користувачів та прохання авторизуватись або зареєструватись
Стан після проведення випробувань	Відображена сторінка сайту «Топ 7 пропозицій»

Таблиця 5.7 – Перевірка відображення сторінки «Ресторани» для неавторизованого користувача

Мета тесту	Перевірка відображення сторінки «Ресторани»
Початковий стан	Відкрита «Головна» сторінка сайту
Схема проведення тесту	Користувач натискає на пункт меню «Ресторани»
Очікуваний результат	Відображення сторінки з пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка пропозиціями від ресторанів



Таблиця 5.8 – Перевірка функції переглянути меню ресторану для неавторизованого користувача

Мета тесту	Перевірка функції переглянути меню ресторану для неавторизованого користувача
Початковий стан	Відкрита сторінка сайту «Ресторани»
Схема проведення тесту	Користувач натискає кнопку «Меню» до бажаної публікації
Очікуваний результат	Відображення попередження, що функція доступна для авторизованих користувачів та прохання авторизуватись або зареєструватись
Стан після проведення випробувань	Відображена сторінка сайту «Ресторани»

Таблиця 5.9 – Перевірка функції переглянути відгуки ресторану для неавторизованого користувача

Мета тесту	Перевірка функції переглянути відгуки ресторану для неавторизованого користувача
Початковий стан	Відкрита сторінка сайту «Ресторани»
Схема проведення тесту	Користувач натискає кнопку «Відгуки» до бажаної публікації
Очікуваний результат	Відображення попередження, що функція доступна для авторизованих користувачів та прохання авторизуватись або зареєструватись
Стан після проведення випробувань	Відображена сторінка сайту «Ресторани»

Таблиця 5.10 – Перевірка відображення сторінки «Кухня» для неавторизованого користувача

Мета тесту	Перевірка відображення сторінки «Кухня»
Початковий стан	Відкрита «Головна» сторінка сайту
Схема проведення тесту	Користувач натискає на пункт меню «Кухня»
Очікуваний результат	Відображення сторінки з кухнями всіх ресторанів міста
Стан після проведення випробувань	Відображення сторінки з кухнями всіх ресторанів міста

Таблиця 5.11 – Перевірка функції додавання кухні до списку уподобань для неавторизованого користувача

Мета тесту	Перевірка функції додавання кухні до списку уподобань для неавторизованого користувача
Початковий стан	Відкрита сторінка сайту «Кухня»
Схема проведення тесту	Користувач натискає кнопку «♥» біля зображення бажаної кухні
Очікуваний результат	Відображення попередження, що функція доступна для авторизованих користувачів та прохання авторизуватись або зареєструватись
Стан після проведення випробувань	Відображена сторінка сайту «Кухня»

Таблиця 5.12 – Перевірка авторизації зареєстрованого користувача до системи

Мета тесту	Перевірка функції авторизації в систему
Початковий стан	Відкрита «Головна» сторінка сайту

Продовження таблиці 5.12

Мета тесту	Перевірка функції авторизації в систему
Вхідні дані	Введення коректного e-mail та паролю до облікового запису користувача, зареєстрованого в системі
Схема проведення тесту	Натиснувши кнопку «Логін», користувач вводить e-mail та пароль до свого особистого кабінету. Після введення необхідних даних користувач натискає кнопку «Увійти»
Очікуваний результат	Відображення сторінки «Головна» з розширеним навігаційним меню
Стан після проведення випробувань	Відображена головна сторінка з розширеним навігаційним меню

Таблиця 5.13 – Перевірка заповнення всіх полів для авторизації

Мета тесту	Перевірка заповнення всіх полів для авторизації
Початковий стан	Відкрита «Головна» сторінка сайту
Вхідні дані	Введення e-mail або паролю до облікового запису користувача
Схема проведення тесту	Натиснувши кнопку «Логін», користувач вводить e-mail або пароль до свого особистого кабінету. Після введення даних користувач натискає кнопку «Увійти»
Очікуваний результат	Відображення вікна з повідомленням про помилку вводу даних для авторизації. Помилка – не заповнене обов'язкове поле «e-mail» або «пароль» Авторизацію не виконано
Стан після проведення випробувань	Відображене вікно для авторизації

Таблиця 5.14 – Перевірка коректності введених даних для авторизації

Мета тесту	Перевірка коректності введених даних для авторизації
Початковий стан	Відкрита «Головна» сторінка сайту
Вхідні дані	Введення e-mail та паролю до облікового запису користувача
Схема проведення тесту	Натиснувши кнопку «Логін», користувач вводить e-mail та пароль з помилкою до свого особистого кабінету. Після введення даних користувач натискає кнопку «Увійти»
Очікуваний результат	Відображення вікна з повідомленням про помилку вводу даних для авторизації. Помилка – поле «e-mail» або «пароль» заповнені не коректними даними Авторизацію не виконано
Стан після проведення випробувань	Відображене вікно для авторизації

Таблиця 5.15 – Перевірка функції відновлення пароля

Мета тесту	Перевірка функції відновлення пароля
Початковий стан	Відкрита «Головна» сторінка сайту
Вхідні дані	Введення e-mail до облікового запису користувача
Схема проведення тесту	Натиснувши кнопку «Логін», користувач вводить e-mail до свого особистого кабінету. Після введення даних користувач натискає кнопку «Відновити пароль»
Очікуваний результат	Відображення вікна з повідомленням про відновлення паролю. «Новий пароль буде відправлено на вказану пошту»
Стан після проведення випробувань	Відображене вікно для авторизації

Таблиця 5.16 – Перевірка функції реєстрації в системі

Мета тесту	Перевірка функції реєстрації в системі
Початковий стан	Відкрита «Головна» сторінка сайту
Вхідні дані	Введення власного ім'я, e-mail та пароля для створення облікового запису користувача
Схема проведення тесту	Натиснувши кнопку «Зареєструватись», користувач вводить своє ім'я, e-mail та пароль для створення особистого кабінету. Після введення даних користувач натискає кнопку «Зареєструватись»
Очікуваний результат	Відображення сторінки «Головна» з розширеним навігаційним меню
Стан після проведення випробувань	Відображена головна сторінка з розширеним навігаційним меню

Таблиця 5.17 – Перевірка відображення сторінки «Рекомендовано мені» для авторизованого користувача

Мета тесту	Перевірка відображення сторінки «Рекомендовано мені»
Початковий стан	Відкрита «Головна» сторінка сайту
Схема проведення тесту	Користувач натискає на пункт меню «Рекомендовано мені»
Очікуваний результат	Відображення сторінки з персональними рекомендаціями для користувача у вигляді публікацій з пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка з персональними рекомендаціями для користувача

Таблиця 5.18 – Перевірка функції «Залишити відгук» для авторизованого користувача

Мета тесту	Перевірка функції «Залишити відгук» для авторизованого користувача
Початковий стан	Відкрита сторінка сайту «Рекомендовано мені»
Вхідні дані	Текст відгуку та оцінка від 1 до 5
Схема проведення тесту	Користувач натискає кнопку «Залишити відгук» до бажаної публікації. У відкритій сторінці з формою для відгуку користувач вводить відгук та виставляє оцінку пропозиції та натискає кнопку «Опублікувати»
Очікуваний результат	Відображення повідомлення про успішну публікацію відгуку
Стан після проведення випробувань	Відображена сторінка «Рекомендовано мені»

Таблиця 5.19 – Перевірка функції додавання страви до списку уподобань для авторизованого користувача

Мета тесту	Перевірка функції додавання страви до списку уподобань для авторизованого користувача
Початковий стан	Відкрита сторінка сайту «Рекомендовано мені»
Схема проведення тесту	Користувач натискає кнопку «♥» біля зображення бажаної страви

Продовження таблиці 5.19

Мета тесту	Перевірка функції додавання страви до списку уподобань для авторизованого користувача
Очікуваний результат	Збереження пропозиції зі стравою в список уподобань в особистому кабінеті користувача. Кнопка «♥» біля обраної публікації змінює свій зовнішній вигляд. Страву успішно додано до списку уподобань
Стан після проведення випробувань	Обрана пропозиція зі стравою збережена в список уподобань в особистому кабінеті користувача. Кнопка «♥» біля обраної публікації змінює свій зовнішній вигляд. Відображення сторінки «Рекомендовано мені»

Таблиця 5.20 – Перевірка функції збереження страви до списку уподобань користувача

Мета тесту	Перевірка функції збереження страви до списку уподобань користувача
Початковий стан	Відкрита сторінка сайту «Рекомендовано мені»
Вхідні дані	Натиснута кнопка «♥» на публікації зі стравою
Схема проведення тесту	Користувач натискає кнопку «♥» біля зображення бажаної страви. Далі натискає на пункт меню навігаційної панелі «Мої уподобання»

Продовження таблиці 5.20

Мета тесту	Перевірка функції збереження страви до списку уподобань користувача
Очікуваний результат	Публікація зі стравою, яку користувач додав у список своїх уподобань відображається на сторінці «Мої уподобання»
Стан після проведення випробувань	Відкрита сторінка «Мої уподобання» зі збереженою стравою

Таблиця 5.21 – Перевірка переходу зі сторінки з уподобаннями до сторінки з ресторанами

Мета тесту	Перевірка переходу на сторінку «Ресторани»
Початковий стан	Відкрита сторінка сайту «Мої уподобання»
Схема проведення тесту	Користувач натискає на пункт меню навігаційної панелі «Ресторани»
Очікуваний результат	Відображення сторінки з пропозиціями від ресторанів
Стан після проведення випробувань	Відображена сторінка пропозиціями від ресторанів

Таблиця 5.22 – Перевірка функції переглянути меню ресторану для авторизованого користувача

Мета тесту	Перевірка функції переглянути меню ресторану для авторизованого користувача
Початковий стан	Відкрита сторінка сайту «Ресторани»
Схема проведення тесту	Користувач натискає кнопку «Меню» до бажаної публікації



Продовження таблиці 5.22

Мета тесту	Перевірка функції переглянути меню ресторану для авторизованого користувача
Очікуваний результат	Відображення сторінки з публікаціями страв обраного ресторану
Стан після проведення випробувань	Відображена сторінка з публікаціями страв обраного ресторану

Таблиця 5.23 – Перевірка функції перегляду відгуків ресторану для авторизованого користувача

Мета тесту	Перевірка функції перегляду відгуків ресторану для авторизованого користувача
Початковий стан	Відкрита сторінка сайту «Ресторани»
Схема проведення тесту	Користувач натискає кнопку «Відгуки» до бажаної публікації
Очікуваний результат	Відображення сторінки з опублікованими відгуками до обраного ресторану
Стан після проведення випробувань	Відображена сторінка сайту з відгуками про ресторан


Таблиця 5.24 – Перевірка відображення сторінки «Кухня» для авторизованого користувача

Мета тесту	Перевірка відображення сторінки «Кухня»
Початковий стан	Відкрита сторінка сайту «Ресторани»
Схема проведення тесту	Користувач натискає на пункт меню «Кухня»
Очікуваний результат	Відображення сторінки з кухнями всіх ресторанів міста
Стан після проведення випробувань	Відображення сторінки з кухнями всіх ресторанів міста

Таблиця 5.25 – Перевірка функції додавання кухні до списку уподобань для авторизованого користувача

Мета тесту	Перевірка функції додавання кухні до списку уподобань для авторизованого користувача
Початковий стан	Відкрита сторінка сайту «Кухня»
Схема проведення тесту	Користувач натискає кнопку «♥» біля зображення бажаної кухні
Очікуваний результат	Збереження обраної кухні в список уподобань в особистому кабінеті користувача. Кнопка «♥» біля обраної кухні змінює свій зовнішній вигляд. Кухню успішно додано до списку уподобань
Стан після проведення випробувань	Обрана кухня збережена в список уподобань в особистому кабінеті користувача. Кнопка «♥» біля обраної кухні змінює свій зовнішній вигляд. Відображення сторінки «Кухня»


Таблиця 5.26 – Перевірка відображення уподобань користувача в особистому кабінеті користувача

Мета тесту	Перевірка відображення уподобань користувача в особистому кабінеті користувача
Початковий стан	Відкрита сторінка сайту «Кухня»
Схема проведення тесту	Користувач натискає на пункт меню особистого кабінету навігаційної панелі у вигляді зображення 

Продовження таблиці 5.26

Мета тесту	Перевірка відображення уподобань користувача в особистому кабінеті користувача
Очікуваний результат	Відображення сторінки особистого кабінету зі збереженими уподобаннями користувача
Стан після проведення випробувань	Відкрита сторінка особистого кабінету зі збереженими уподобаннями користувача

Таблиця 5.27 – Перевірка функції зміни паролю користувача

Мета тесту	Перевірка функції зміни паролю користувача
Початковий стан	Відкрита сторінка особистого кабінету користувача
Вхідні дані	Новий пароль до облікового запису
Схема проведення тесту	Користувач натискає на кнопку  біля відповідного поля зміни пароля. Вводить новий пароль та натискає кнопку Enter
Очікуваний результат	Відображення повідомлення про успішну зміну пароля. Оновлений пароль зберігається в обліковому записі користувача
Стан після проведення випробувань	Відкрита сторінка особистого кабінету з оновленим паролем

Таблиця 5.28 – Перевірка функції виходу з особистого кабінету користувача

Мета тесту	Перевірка функції виходу з особистого кабінету користувача
Початковий стан	Відкрита сторінка особистого кабінету користувача
Схема проведення тесту	Користувач натискає на кнопку «Вийти»
Очікуваний результат	Користувач виходить з особистого кабінету та потрапляє на головну сторінку сайту
Стан після проведення випробувань	Відкрита головна сторінка сайту

Таблиця 5.29 – Перевірка авторизації адміністратора до системи

Мета тесту	Перевірка функції авторизації адміністратора в систему
Початковий стан	Відкрита «Головна» сторінка сайту
Вхідні дані	Введення коректного e-mail та паролю адміністратора, зареєстрованого в системі
Схема проведення тесту	Натиснувши кнопку «Логін», користувач вводить e-mail та пароль до свого особистого кабінету адміністратора. Після введення необхідних даних користувач натискає кнопку «Увійти»
Очікуваний результат	Відображення сторінки особистого кабінету адміністратора
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

Таблиця 5.30 – Перевірка функції додавання нового ресторану адміністратором

Мета тесту	Перевірка функції додавання нового ресторану адміністратором
Початковий стан	Відкрита сторінка особистого кабінету адміністратора
Вхідні дані	Введення необхідних даних для додавання нового ресторану: назва ресторану, адреса, район, контактний номер, веб-сайт та зображення ресторану.
Схема проведення тесту	Адміністратор натискає кнопку «Додати новий» на панелі керування ресторанами. Далі адміністратор вводить необхідні дані про ресторан у відповідні поля, додає зображення ресторану шляхом «перетягування» зображення у відповідне віконце для зображення. Натискає кнопку «Додати ресторан».
Очікуваний результат	Відображення повідомлення про успішне додавання нового ресторану в систему
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

Таблиця 5.31 – Перевірка функції оновлення даних про страву адміністратором

Мета тесту	Перевірка функції оновлення даних про страву адміністратором
Початковий стан	Відкрита сторінка особистого кабінету адміністратора
Вхідні дані	Введення оновлених даних для страви: назва страви, опис, кухня, ресторан, ціна або зображення страви.

Продовження таблиці 5.31

Мета тесту	Перевірка функції оновлення даних про страву адміністратором
Схема проведення тесту	Адміністратор натискає кнопку «Редагувати існуючу» на панелі керування стравами. У відкритій сторінці з усіма існуючими в системі стравами, адміністратор обирає потрібну страву для редагування та натискає кнопку «Редагувати». Далі у відкритому вікні для редагування адміністратор вводить необхідні дані про страву у відповідні поля, оновлює зображення страви шляхом «перетягування» зображення у відповідне віконце для зображення. Натискає кнопку «Оновити страву».
Очікуваний результат	Відображення повідомлення про успішне оновлення інформації про страву
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

Таблиця 5.32 – Перевірка функції видалення страви адміністратором

Мета тесту	Перевірка функції видалення страви адміністратором
Початковий стан	Відкрита сторінка особистого кабінету адміністратора
Схема проведення тесту	Адміністратор натискає кнопку «Редагувати існуючу» на панелі керування стравами. У відкритій сторінці з усіма існуючими в системі стравами, адміністратор обирає потрібну страву для видалення та натискає кнопку «Видалити».

Продовження таблиці 5.32

Мета тесту	Перевірка функції видалення страви адміністратором
Очікуваний результат	Відображення повідомлення про успішне видалення страви
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

Таблиця 5.33 – Перевірка функції призначення нового адміністратора

Мета тесту	Перевірка функції призначення нового адміністратора
Початковий стан	Відкрита сторінка особистого кабінету адміністратора
Схема проведення тесту	Адміністратор натискає кнопку «Призначити адміном» на панелі керування користувачами. У відкритій сторінці з усіма зареєстрованими користувачами, адміністратор обирає користувача та натискає кнопку «Призначити адміном».
Очікуваний результат	Відображення повідомлення про успішне призначення нового адміністратора системи
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

Таблиця 5.34 – Перевірка функції видалення користувача

Мета тесту	Перевірка функції видалення користувача
Початковий стан	Відкрита сторінка особистого кабінету адміністратора
Схема проведення тесту	Адміністратор натискає кнопку «Видалити» на панелі керування користувачами. У відкритій сторінці з усіма зареєстрованими користувачами, адміністратор обирає користувача та натискає кнопку «Видалити юзера».
Очікуваний результат	Відображення повідомлення про успішне видалення користувача
Стан після проведення випробувань	Відображена сторінка особистого кабінету адміністратора

### Висновок до розділу

У цьому розділі були описані випробування програмного продукту, тобто проведення тестів та порядок їх виконання для перевірки відповідності розробленого програмного продукту функціональним вимогам, представленим у технічному завданні на створення інформаційної системи пошуку ресторанів.

Перш за все було визначено мету випробувань: перевірити відповідність функцій інформаційної системи пошуку ресторанів вимогам, вказаним у технічному завданні.

Результати випробувань представлено у вигляді таблиць.



## ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання дипломного проекту були ґрунтовно розглянуті питання доречності та необхідності створення інформаційної системи пошуку ресторанів.

Був проведений докладний аналіз предметного середовища, детально описаний процес формування персональної пропозиції.

На основі даних, отриманих у процесі аналізу, було сформульовано відповідні математичні задачі. Розглянуто ключові етапи формування персональної рекомендації та розроблено алгоритм за допомогою математичних методів розв’язання.

Було представлено опис функціональної моделі, визначені ключові актори та основні дії акторів у системі. Спираючись на розроблені варіанти використання були виявлені та реалізовані функціональні вимоги системи.

Для розробки програмного забезпечення було використано наступні засоби: TypeScript, Node.js, EJS, Express.js, для розробки клієнтської частини — HTML, CSS, JS, Bootstrap.

Для роботи з даними, спочатку була розроблена модель бази даних, яка надає можливість ефективно та надійно здійснювати доступ до даних. Була обрана NoSQL база даних MongoDB у парі з системою управління базою Redis. Для розгортання продукту був використаний Docker.

Для представлення архітектури програмного забезпечення побудовано діаграми класів, послідовності та компонентів з детальним описом цих діаграм. Були наведені специфікації функцій, які задіяні в роботі програми задля формування рекомендацій користувачеві.

Розроблена інструкція для користувача по експлуатації системи. Перевірені системні функції, що виконуються системою. Детально описана методика проведення випробувань, яка показує можливість введення програми в експлуатацію.

						Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ПОСИЛАНЬ

- 1) API [Електронний ресурс] – Режим доступу URL:  
<https://habr.com/ru/sandbox/52599/>
- 2) Колаборативна фільтрація [Електронний ресурс] – Режим доступу URL:  
<https://cyberleninka.ru/article/n/kollaborativnaya-filtratsiya-dlya-postroeniya-rekomendatsiy-na-osnove-dannyh-o-zakazah/viewer>
- 3) Node.js [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/Node.js>
- 4) Bootstrap [Електронний ресурс] – Режим доступу URL:  
<https://getbootstrap.com/>
- 5) Алгоритм колаборативної фільтрації [Електронний ресурс] – Режим доступу URL:  
<https://habr.com/ru/post/80955/>
- 6) Item based колаборативна фільтрація [Електронний ресурс] – Режим доступу URL:  
<https://habr.com/ru/company/ivi/blog/232843/>
- 7) Tripadvisor [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/TripAdvisor>
- 8) Yelp [Електронний ресурс] – Режим доступу URL:  
<https://www.yelp.com/>
- 9) TypeScript [Електронний ресурс] – Режим доступу URL: <https://uk.wikipedia.org/wiki/TypeScript>
- 10) Express.js [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/Express.js>
- 11) EJS [Електронний ресурс] – Режим доступу URL:  
<http://docs.mirocow.com/doku.php?id=nodejs:ejs>.
- 12) MongoDB [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/MongoDB>.

- 13) Redis [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/Redis>.
- 14) Docker [Електронний ресурс] – Режим доступу URL:  
<https://docs.docker.com/engine/>.
- 15) Mustache.js [Електронний ресурс] – Режим доступу URL:  
<https://sitear.ru/material/mustache-js-razrabotka-html-shablonov-s-mustache>.

16) CUSTOMER'S DEMAND ESTIMATION  
 WITH THE USE OF COLLABORATIVE FILTRATION, E.G. Tsilova, E.J.  
 Ekgaus, с. 89-94

- 17) Макарь В.А. Колаборативна фільтрація в інформаційних  
 системах. [Електронний ресурс] – Режим доступу URL:  
[http://sociosphaera.com/publication/conference/2012/160kollaborativnaya\\_filt\\_raciya\\_v\\_informacionnyh\\_sistemah](http://sociosphaera.com/publication/conference/2012/160kollaborativnaya_filt_raciya_v_informacionnyh_sistemah)
- 18) Apache Mahout. Метрики для визначення схожості користувачів.  
 [Електронний ресурс] – Режим доступу URL:  
<http://habrahabr.ru/post/188350/>.
- 19) JavaScript [Електронний ресурс] – Режим доступу URL:  
<https://uk.wikipedia.org/wiki/JavaScript>
- 20) Казимир В. В., Пріла О. А., Крищенко М. С., (2017) ЗАСТОСУВАННЯ  
 ДИНАМІЧНИХ ВІРТУАЛЬНИХ ОБРАЗІВ У ГРІД-СЕРЕДОВИЩІ З  
 ПІДТРИМКОЮ РЕПЛІКАЦІЇ. [Електронний ресурс] – Режим доступу  
 URL: <https://tst.stu.cn.ua/tmppdf/297.pdf>

## Додаток А

*Тексти програмного коду**Інформаційна система пошуку ресторанів*

(Найменування програми (документа))

*DVD-R*

(Вид носія даних)

*16 арк, 80 Мб*

(Обсяг програми (документа) , арк.,) Кб)

Київ – 2020 року

Змн.	Арк.	№ докум.	Підпис	Дата

*Лістинг файлу index.ts — ініціалізація та налаштування всього застосунку*

```
import express from "express";
import session from "express-session";
import bodyParser from "body-parser";
import passport from "passport";

import * as dishController from "controllers/dishController";
import * as kitchenController from "controllers/kitchenController";
import * as restaurantController from "controllers/restaurantController";
import * as reviewController from "controllers/reviewController";
import { Schema } from "mongoose";

require("configs/database");
require("configs/passport");
const app = express();
app.set("view engine", "ejs");
app.use(express.static("public"));

app.use(session({
  secret: "restik",
  resave: false,
  saveUninitialized: false,
}));

app.use(bodyParser.urlencoded({ extended: true }));
app.use(passport.initialize());
app.use(passport.session());

app.post("/login", passport.authenticate("local", { successRedirect: "/" }));
app.get("/", (req, res) => {
```

Змн.	Арк.	№ докум.	Підпис	Дата

```
res.render("index", {  
  
  user: req.user  
  
});  
  
});  
  
app.get("/proposals", (req, res) => {  
  
  dishController.getTopDishes((err, dishes) => {  
  
    res.render("proposals", {  
  
      dishes  
  
    });  
  
  });  
  
});  
  
app.get("/kitchen", (req, res) => {  
  
  kitchenController.getKitchens((err, kitchens) => {  
  
    res.render("kitchen", {  
  
      kitchens  
  
    });  
  
  });  
  
});  
  
app.get("/restaurants", (req, res) => {  
  
  restaurantController.getRestaurants((err, restaurants) => {  
  
    res.render("restaurants", {  
  
      restaurants
```

Змн.	Арк.	№ докум.	Підпис	Дата

});

})

});

app.get("/review/:restaurantId", (req, res) =&gt; {

const restaurantId = new Schema.Types.ObjectId(req.params.restaurantId);

reviewController.findReviewsByRestaurant(restaurantId, (err, reviews) =&gt; {

res.render("review", {

reviews

});

});

});

app.listen(3000);

*Лістинг файлу dish.ts — модель страви*

import mongoose, { Document, Schema } from "mongoose";

import { KitchenDocument } from "models/kitchen";

import { RestaurantDocument } from "models/restaurant";

export interface DishDocument extends Document {

name: string;

description: string;

price: number;

kitchen: KitchenDocument;

restaurant: RestaurantDocument;

```

    imageUrl: string;

  }

  const DishSchema = new Schema<DishDocument>({

    name: String,

    description: String,

    price: Number,

    kitchen: { type: Schema.Types.ObjectId, ref: "Kitchen" },

    restaurant: { type: Schema.Types.ObjectId, ref: "Restaurant" },

    imageUrl: String

  });

  export const Dish = mongoose.model<DishDocument>("Dish", DishSchema);

```

*Лістинг файлу kitchen.ts — модель кухні*

```

import mongoose, { Document, Schema } from "mongoose";

export interface KitchenDocument extends Document {

  name: string;

  imageUrl: string;

}

const KitchenSchema = new Schema<KitchenDocument>({

  name: String,

  imageUrl: String

});

```

Змн.	Арк.	№ докум.	Підпис	Дата



```
export const Kitchen = mongoose.model<KitchenDocument>("Kitchen",
KitchenSchema);
```

*Листинг файлу restaurant.ts — модель ресторану*

```
import mongoose, { Document, Schema } from "mongoose";
```

```
export interface RestaurantDocument extends Document {
```

```
  name: string;
```

```
  address: string;
```

```
  district: string;
```

```
  phone: string;
```

```
  website: string;
```

```
  imageUrl: string;
```

```
}
```

```
const RestaurantSchema = new Schema<RestaurantDocument>({
```

```
  name: String,
```

```
  address: String,
```

```
  district: String,
```

```
  phone: String,
```

```
  website: String,
```

```
  imageUrl: String
```

```
});
```

```
export const Restaurant = mongoose.model<RestaurantDocument>("Restaurant",
RestaurantSchema);
```

*Лістинг файлу review.ts — модель відгуку*

```
import mongoose, { Document, Schema } from "mongoose";
import { UserDocument } from "models/user";
import { DishDocument } from "models/dish";
export interface ReviewDocument extends Document {
  user: UserDocument;
  dish: DishDocument;
  rating: number;
  text: string;
}
const ReviewSchema = new Schema<ReviewDocument>({
  user: { type: Schema.Types.ObjectId, ref: "User" },
  dish: { type: Schema.Types.ObjectId, ref: "Dish" },
  rating: Number,
  text: String
});
export const Review = mongoose.model<ReviewDocument>("Review",
ReviewSchema);
```

*Лістинг файлу user.ts — модель користувача (адміністратора)*

```
import mongoose, { Document, Schema } from "mongoose";
export interface UserDocument extends Document {
  name: string;
  email: string;
  password: string;
  isAdmin: boolean;
}
const UserSchema = new Schema<UserDocument>({
```

name: String,

email: String,

password: String,

isAdmin: Boolean

});

export const User = mongoose.model<UserDocument>("User", UserSchema);

*Лістинг файлу dishController.ts — контролер страв*

import { Dish, DishDocument } from "models/dish";

import { Schema } from "mongoose";

import Raccoon from "@maruware/raccoon";

export function getTopDishes(callback: (err: any, res: DishDocument[]) => void) {

Dish.find().sort("likes").exec(callback);

}

export function findDishesByName(name: string, callback: (err: any, res:

DishDocument[]) => void) {

Dish.\$where(`this.name == "\${name}"`).exec(callback);

}

export function findDishesByKitchen(kitchen: Schema.Types.ObjectId, callback:

(err: any, res: DishDocument[]) => void) {

Dish.\$where(`this.kitchen == "\${kitchen}"`).exec(callback);

}

```
export async function getCollaborativeRecommendations(userId: string, callback:
(err: any, res: DishDocument[]) => void) {

    const raccoon = new Raccoon({ });

    const dishIds = await raccoon.recommendFor(userId, 10);

    Dish.find().where('id').in(dishIds).exec(callback);

}
```

*Лістинг файлу kitchenController.ts — контролер кухонь*

```
import { Kitchen, KitchenDocument } from "models/kitchen";

export function getKitchens(callback: (err: any, res: KitchenDocument[]) => void)
{

    return Kitchen.find(callback);

}
```

*Лістинг файлу restaurantController.ts — контролер ресторанів*

```
import { Restaurant, RestaurantDocument } from "models/restaurant";

export function getRestaurants(callback: (err: any, res: RestaurantDocument[]) =>
void) {

    return Restaurant.find(callback);

}
```

*Лістинг файлу userController.ts — контролер користувачів*

```
import { User, UserDocument } from "models/user";

export function getUsers(callback: (err: any, res: UserDocument[]) => void) {
```

					ДП 6225.00.000 ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    return User.find(callback);

}

export function findUserByEmail(email: string, callback: (err: any, res:
UserDocument[]) => void) {

    return User.$where(`this.email == '${email}'`).find(callback);

}

```

*Лістинг файлу reviewController.ts — контролер відгуків*

```

import { Review, ReviewDocument } from "models/review";

import { Schema } from "mongoose";

export function findReviewsByRestaurant(restaurant: Schema.Types.ObjectId,
callback: (err: any, res: ReviewDocument[]) => void) {

    Review.$where(`this.dish.restaurant == '${restaurant}'`).find(callback);

}

```

*Лістинг файлу docker-compose.yml — налаштування зв'язку та відкриття портів між сервісами*

```

version: '3.7'

services:

mongo:

    image: mongo

    restart: always

    ports:

```

Змн.	Арк.	№ докум.	Підпис	Дата

- 27017:27017

environment:

MONGO\_INITDB\_ROOT\_USERNAME: root

MONGO\_INITDB\_ROOT\_PASSWORD: root

mongo-express:

image: mongo-express

restart: always

ports:

- 8081:8081

environment:

ME\_CONFIG\_MONGODB\_ADMINUSERNAME: root

ME\_CONFIG\_MONGODB\_ADMINPASSWORD: root

*Лістинг файлу package.json — налаштування зв'язку для обміну пакетами*

```
{
  "name": "restik",
  "private": true,
  "scripts": {
    "watch": "tsc-watch --onSuccess \"node dist/index.js\""
  },
  "dependencies": {
    "ejs": "^3.1.3",
    "express": "^4.17.1",
    "express-session": "^1.17.1",
    "mongoose": "^5.9.17",
    "passport": "^0.4.1",
    "passport-local": "^1.0.0"
  },
  "devDependencies": {
    "@types/express": "^4.17.6",
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

"@types/express-session": "^1.17.0",
"@types/mongoose": "^5.7.22",
"@types/passport": "^1.0.3",
"@types/passport-local": "^1.0.33",
"nodemon": "^2.0.4",
"tsc-watch": "^4.2.8",
"typescript": "^3.9.5"
}
}

```

*Лістинг файлу index.ejs — представлення головної сторінки*

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
  integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYYxFfc+NcPb1dK
Gj7Sk" crossorigin="anonymous">
  <link rel="stylesheet" href="/assets/css/main.css">

  <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@500&display=swa
p" rel="stylesheet">
</head>

body>

```

&lt;!-- Логина --&gt;

&lt;form action="" id="login"&gt;

&lt;span&gt;E-mail &lt;/span&gt;

&lt;input type="email"&gt;

&lt;span&gt;Пароль&lt;/span&gt;

&lt;input type="password"&gt;

&lt;button type="submit"&gt;Увійти&lt;/button&gt;

&lt;a href="#" onclick="resetPass()"&gt;Відновити Пароль&lt;/a&gt;

&lt;span

class="close"

onclick="document.getElementById('login').style.display='none'"

title="Close Modal"&gt;&amp;times;&lt;/span&gt;

&lt;/form&gt;

&lt;!-- Форма регистрации --&gt;

&lt;form action="" id="register"&gt;

&lt;span&gt;Ваше ім'я&lt;/span&gt;

&lt;input type="text"&gt;

&lt;span&gt;E-mail &lt;/span&gt;

&lt;input type="email"&gt;

&lt;span&gt;Пароль&lt;/span&gt;

&lt;input type="password"&gt;

&lt;button type="submit"&gt;Зареєструватись&lt;/button&gt;

&lt;span

class="close"

onclick="document.getElementById('register').style.display='none'"

title="Close Modal"&gt;&amp;times;&lt;/span&gt;

&lt;/form&gt;

&lt;!-- Восстановить пароль--&gt;

&lt;div class="notlogged" id="resetPass"&gt;

&lt;p style="font-size: 21px;"&gt;Новий пароль буде відправлено на:&lt;/p&gt;

Змн.	Арк.	№ докум.	Підпис	Дата



```

<input type="text" placeholder="example@gmail.com" class="form-control">
<button class="form-control"
onclick="document.getElementById('resetPass').style.display='none'">Добре</but
ton>

<span class="close"
onclick="document.getElementById('resetPass').style.display='none'"
title="Close Modal">&times;</span>
</div>

<nav class="navbar navbar-expand-lg navbar-light bg-transparent">
  <a class="logo navbar-brand" href="/" style="color: white;">RESTIK <br>
  FINDER</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
  target="#navbarNav"
    aria-controls="navbarNav"    aria-expanded="false"    aria-label="Toggle
  navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav">
      <li class="nav-item ">
        <a class="nav-link active" href="/">Головна </a>
      </li>
      <li class="nav-item ">
        <a class="nav-link" href="/proposals">Топ 7 пропозицій</a>
      </li>
      <% if (user) { %>
      <li class="nav-item">
        <a class="nav-link" href="/recommended">Рекомендовано мені</a>
      </li>

```

Змн.	Арк.	№ докум.	Підпис	Дата

&lt;% } %&gt;

&lt;li class="nav-item "&gt;

&lt;a class="nav-link " href="/restaurants"&gt;Ресторани&lt;/a&gt;

&lt;/li&gt;

&lt;li class="nav-item"&gt;

&lt;a class="nav-link" href="/kitchen"&gt;Кухня&lt;/a&gt;

&lt;/li&gt;

&lt;% if (user) { %&gt;

&lt;li class="nav-item"&gt;

&lt;a class="nav-link" href="/liked"&gt;Мої уподобання&lt;/a&gt;

&lt;/li&gt;

&lt;li class="nav-item"&gt;

<a class="nav-link" href="/admin"></a>

&lt;/li&gt;

&lt;% } else { %&gt;

&lt;li class="nav-item login"&gt;

&lt;a class="nav-link " href="#"

onclick="event.preventDefault();

document.getElementById('login').style.display='flex'&gt;

Логін

&lt;/a&gt;

&lt;hr&gt;

&lt;a class="nav-link " href="#"

onclick="event.preventDefault();

document.getElementById('register').style.display='flex'&gt;

Зареєструватись

&lt;/a&gt;

&lt;/li&gt;

Змн.	Арк.	№ докум.	Підпис	Дата

&lt;% } %&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/nav&gt;

&lt;div class="container"&gt;

&lt;div class="search-rest"&gt;

<h2>Знайди свій улюблений <span style="color:red;">RESTIK!</span></h2>

&lt;div class="search-group"&gt;

&lt;div class="row"&gt;

&lt;div class="col"&gt;

<input type="text" class="form-control" placeholder="Введіть кухню">

&lt;/div&gt;

&lt;div class="col"&gt;

&lt;select class="form-control custom-select"&gt;

&lt;option selected&gt;Виберіть район&lt;/option&gt;

&lt;option value="1"&gt;Деснянский район&lt;/option&gt;

&lt;option value="2"&gt;Днепровский район&lt;/option&gt;

&lt;option value="3"&gt;Оболонский район&lt;/option&gt;

&lt;option value="3"&gt;Дарницкий район&lt;/option&gt;

&lt;option value="3"&gt;Голосеевский район&lt;/option&gt;

&lt;option value="3"&gt;Соломенский район&lt;/option&gt;

&lt;option value="3"&gt;Подольский район&lt;/option&gt;

&lt;option value="3"&gt;Шевченковский район&lt;/option&gt;

&lt;option value="3"&gt;Центр&lt;/option&gt;

&lt;/select&gt;

&lt;/div&gt;

&lt;div class="col"&gt;

					ДП 6225.00.000 ПЗ	Арк. 78
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <button          type="submit"          class="search-icon"></button>

```

```

    </div>

```

```

  </div>

```

```

</div>

```

```

</div>

```

```

</div>

```

```

<script src="/assets/js/script.js"></script>

```

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

```

```

    integrity="sha384-

```

```

DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa
Rkfj"

```

```

    crossorigin="anonymous"></script>

```

```

<script

```

```

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

```

```

    integrity="sha384-

```

```

Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfoo
Ao"

```

```

    crossorigin="anonymous"></script>

```

```

<script

```

```

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"

```

```

    integrity="sha384-

```

```

OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0J
KI"

```

```

    crossorigin="anonymous"></script>

```

```

</body>

```

```

function Cuisine() {

```

					ДП 6225.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79

```
document.getElementById('editCuisine').style.display = 'none';

document.getElementById('editElem').style.display = 'flex';

}

function infoMeal(e) {

    document.getElementById('edit' + e).style.display = 'none';

    document.getElementById('info' + e).style.display = 'flex';

}

function meal(e) {

    if (e == 'delete') {

        document.getElementById(e + 'Meals').style.display = 'flex';

    } else {

        document.getElementById('editMeal').style.display = 'none';

        document.getElementById('editMeals').style.display = 'flex';

    }

}

function rest(e) {

    if (e == 'delete') {

        document.getElementById(e + 'Rests').style.display = 'flex';

    } else {

        document.getElementById('editRest').style.display = 'none';

        document.getElementById('editRests').style.display = 'flex';

    }

}

}
```

```

function resetPass() {

    document.getElementById('login').style.display = 'none';

    document.getElementById('resetPass').style.display = 'flex';

}

function likeevent(e) {

    for (let index = 4; index > e.target.id; index--) {

        document.getElementById(index).src = './assets/img/star (1).svg';

    }

    for (let index = 0; index <= e.target.id; index++) {

        document.getElementById(index).src = './assets/img/star.svg';

    }

}

function fixLike(e) {

    for (let index = 0; index < 5; index++) {

        document.getElementById(index).removeEventListener('mousemove',
likeevent)

    }

}

function Like() {

    let p = document.getElementById('rating');

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

for (let index = 0; index < 5; index++) {

    let span = document.createElement('img');

    span.setAttribute('id', index);

    span.setAttribute('style', "width: 20px; height: 20px;");

    span.setAttribute('src', "./assets/img/star (1).svg");

    span.addEventListener('mousemove', likeevent)

    span.addEventListener('click', fixLike)

    p.appendChild(span)

}

}

Like();

```

*Лістинг файлу proposals.ejs — представлення сторінки з пропозиціями*

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

```

					ДП 6225.00.000 ПЗ	Арк. 82
Змн.	Арк.	№ докум.	Підпис	Дата		

integrity="sha384-

9aIt2nRpC12Uk9gS9baDI411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dK  
Gj7Sk" crossorigin="anonymous">

<link rel="stylesheet" href="/assets/css/main.css">

<link rel="stylesheet" href="/assets/css/proposals.css">

<link

href="https://fonts.googleapis.com/css2?family=Roboto:wght@500&display=swap" rel="stylesheet">

</head>

<body>

<!-- Логина -->

<form action="" id="login">

<span>E-mail </span>

<input type="email">

<span>Пароль</span>

<input type="password">

<button type="submit">Увійти</button>

<u onclick="resetPass()">Відновити Пароль</u>

<span class="close"

onclick="document.getElementById('login').style.display='none'"

title="Close Modal">&times;</span>

</form>

					ДП 6225.00.000 ПЗ	Арк.
						83
Змн.	Арк.	№ докум.	Підпис	Дата		



<!-- Форма регистрации -->

<form action="" id="register">

<span>Ваше ім'я</span>

<input type="text">

<span>E-mail </span>

<input type="email">

<span>Пароль</span>

<input type="password">

<button type="submit">Зареєструватись</button>

<span class="close"

onclick="document.getElementById('register').style.display='none'"

title="Close Modal">&times;</span>

</form>

<!-- Восстановить пароль-->

<div class="notlogged" id="resetPass">

<p style="font-size: 21px;">Новий пароль буде відправлено на:</p>

<input type="text" placeholder="example@gmail.com" class="form-control">

<button class="form-control"

onclick="document.getElementById('resetPass').style.display='none'">Добре</button>

Змн.	Арк.	№ докум.	Підпис	Дата

```
<span class="close"
onclick="document.getElementById('resetPass').style.display='none'"
title="Close Modal">&times;</span>

</div>

<nav class="navbar navbar-expand-lg navbar-light bg-transparent">

  <a class="logo navbar-brand" href="/" style="color: white;">RESTIK <br>
  FINDER</a>

  <button class="navbar-toggler" type="button" data-toggle="collapse" data-
  target="#navbarNav"

    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
  navigation">

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarNav">

    <ul class="navbar-nav">

      <li class="nav-item ">

        <a class="nav-link" href="/">Головна </a>

      </li>

      <li class="nav-item ">

        <a class="nav-link active" href="/proposals">Топ 7 пропозицій</a>

      </li>

      <% if (user) { %>
```

```
<li class="nav-item">
```

```
    <a class="nav-link" href="/recommended">Рекомендовано мені</a>
```

```
</li>
```

```
<% } %>
```

```
<li class="nav-item ">
```

```
    <a class="nav-link " href="/restaurants">Ресторани</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
    <a class="nav-link" href="/kitchen">Кухня</a>
```

```
</li>
```

```
<% if (user) { %>
```

```
<li class="nav-item">
```

```
    <a class="nav-link" href="/liked">Мої уподобання</a>
```

```
</li>
```

```
<li class="nav-item">
```

```
    <a class="nav-link" href="/admin"></a>
```

```
</li>
```

```
<% } else { %>
```

```
<li class="nav-item login">
```

```
    <a class="nav-link " href="#"
```

```
        onclick="event.preventDefault();
```

```
        document.getElementById('login').style.display='flex'>
```

Логін

&lt;/a&gt;

&lt;hr&gt;

&lt;a class="nav-link " href="#"

onclick="event.preventDefault();

document.getElementById('register').style.display='flex'"&gt;

Зареєструватись

&lt;/a&gt;

&lt;/li&gt;

&lt;% } %&gt;

&lt;/ul&gt;

&lt;/div&gt;

&lt;/nav&gt;

&lt;div class="container-sort"&gt;

&lt;div class="sort-menu"&gt;

&lt;p&gt;Сортувати за&lt;/p&gt;

&lt;select class="form-control custom-select"&gt;

&lt;option selected&gt;&lt;/option&gt;

&lt;option value="1"&gt;Суши&lt;/option&gt;

&lt;/select&gt;

&lt;p&gt;Розташування &lt;/p&gt;

&lt;select class="form-control custom-select"&gt;

&lt;option selected&gt;&lt;/option&gt;

					ДП 6225.00.000 ПЗ	Арк. 87
Змн.	Арк.	№ докум.	Підпис	Дата		

<option value="1">Деснянський район</option>

</select>

</div>

<div class="restaraunt">

<div class="meal">

<div class="red-box" id="like">



</div>

</div>

<div class="rest-description">

<h4>Здоровий глузд</h4>

<p><span style="color: red;"> ☆ ☆ ☆ ☆ </span> 4,8</p>

<p style="margin: 5px 0;">Карпаччо з яловичини</p>

<p style="font-size: 13px;">Карпаччо з яловичини та маринованим перцем</p>

<span style="color: #6D6D6D; margin-top: 20px;"> Європейська <span

style="margin-left: 10px; color:#FF4C4F;">115

грн</span></span>

</div>

<div class="rest-place">

					ДП 6225.00.000 ПЗ	Арк.
						88
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<div class="rest-place-i">
```

```
<p>address</p>
```

```
<div class="red-box">
```

```

```

```
</div>
```

```
</div>
```

```
<div class="rest-place-i">
```

```
<p>phone</p>
```

```
<div class="red-box">
```

```

```

```
</div>
```

```
</div>
```

```
<button class="btn" onClick='location.href="/review"'>Залишити
```

```
відгук</button>
```

```
</div>
```

```
</div>
```

```
<div class="restaraunt">
```

```
<div class="meal">
```

```
<div class="red-box">
```

```

```

Змн.	Арк.	№ докум.	Підпис	Дата

&lt;/div&gt;

&lt;/div&gt;

&lt;div class="rest-description"&gt;

&lt;h4&gt;Здоровий глузд&lt;/h4&gt;

&lt;p&gt;&lt;span style="color: red;"&gt; ☆ ☆ ☆ ☆ &lt;/span&gt; 4,8&lt;/p&gt;

&lt;p style="margin: 5px 0;"&gt;Карпаччо з яловичини&lt;/p&gt;

&lt;p style="font-size: 13px;"&gt;Карпаччо з яловичини та маринованим перцем&lt;/p&gt;

&lt;span style="color: #6D6D6D; margin-top: 20px;"&gt; Європейська &lt;span style="margin-left: 10px; color: #FF4C4F;"&gt;115 грн&lt;/span&gt;&lt;/span&gt;

&lt;/div&gt;

&lt;div class="rest-place"&gt;

&lt;div class="rest-place-i"&gt;

&lt;p&gt;address&lt;/p&gt;

&lt;div class="red-box"&gt;

&lt;img src="/assets/img/marker.png" /&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;div class="rest-place-i"&gt;

&lt;p&gt;phone&lt;/p&gt;

&lt;div class="red-box"&gt;

&lt;img src="/assets/img/phone.png" alt=""&gt;

					ДП 6225.00.000 ПЗ	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

&lt;/div&gt;

&lt;/div&gt;

<button class="btn" onClick='location.href="/review"'>Залишити  
відгук</button>

&lt;/div&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;/div&gt;

&lt;script src="/assets/js/like.js"&gt;&lt;/script&gt;

&lt;script src="/assets/js/script.js"&gt;&lt;/script&gt;

&lt;script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"

integrity="sha384-

DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa  
Rkfj"

crossorigin="anonymous"&gt;&lt;/script&gt;

&lt;script

src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

integrity="sha384-

Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfoo  
Ao"

crossorigin="anonymous"&gt;&lt;/script&gt;

&lt;script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"

integrity="sha384- /kR0JKI"

					ДП 6225.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		91



```
crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

*Лістинг файлу database.ts — налаштування зовнішньої бібліотеки, підключення до бази даних*

```
import mongoose from 'mongoose';
mongoose.connect('mongodb://localhost:27017/test', { useNewUrlParser: true });
```

*Лістинг файлу passport.ts — налаштування зовнішньої бібліотеки, авторизації користувачів та адміністратора*

```
import passport from "passport"
import { Strategy as LocalStrategy } from "passport-local"
import { User, UserDocument } from "models/user";

passport.use(new LocalStrategy((username, password, done) => {
  User.findOne({ email: username }, (err, res) => {
    done(err, res);
  })
}));

passport.serializeUser<UserDocument, string>((user, done) => {
  done(null, user.id);
});

passport.deserializeUser<UserDocument, string>((id, done) => {
  User.findById(id, (err, res) => {
    done(err, res ?? undefined);
  });
});
```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”  
Кафедра автоматизованих систем обробки інформації і управління

**УЗГОДЖЕНО**

**Керівник проекту**

\_\_\_\_\_ Олесь КОВТУНЕЦЬ

(підпис)

(вл. ім'я, прізвище)

“ 13 ” квітня 2020 р.

**ЗАТВЕРДЖУЮ**

**В.о. завідувача кафедри**

\_\_\_\_\_ О.А. Павлов

(підпис)

(ініціали, прізвище)

“ 14 ” квітня 2020 р.

Інформаційна система пошуку ресторанів

**ТЕХНІЧНЕ ЗАВДАННЯ**

Шифр *ДП 6225.01.1081-с ТЗ*

на 10 сторінках

Київ – 2020 року

## ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1 Повне найменування системи та її умовне позначення.....	3
1.2 Найменування організації-замовника та організацій-учасників робіт.....	3
1.3 Перелік документів, на підставі яких створюється система (Завдання на ДП).....	3
1.4 Планові терміни початку і закінчення роботи зі створення системи.....	3
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ.....	4
2.1 Призначення системи .....	4
2.2 Цілі створення системи.....	4
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	5
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	6
4.1 Вимоги до функціональних характеристик.....	6
4.2 Вимоги до надійності.....	6
4.3 Вимоги до складу і параметрів технічних засобів .....	6
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ.....	8
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	9
6.1 Види випробувань .....	9

					<b>ДП ІС-6225.01.000 ТЗ</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>Прізвище</i>	<i>Підпис</i>	<i>Дата</i>	<i>Інформаційна система пошуку ресторанів</i>	<i>Лім.</i>	<i>Лист</i>	<i>Листів</i>
<i>Розроб.</i>		<i>Федько М.Р.</i>						
<i>Перевірив.</i>		<i>Ковтунець О.В.</i>					2	10
						<i>КПІ ім. ІгоряСікорського кафедра АСОІУ гр. ІС-62</i>		
<i>Н. кон.</i>		<i>Новінський В.П.</i>						
<i>Затв.</i>		<i>Павлов О.А.</i>						

## 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

### 1.1 Повне найменування системи та її умовне позначення

**Повна назва системи:** Інформаційна система пошуку ресторанів

**Коротке найменування системи:** «RESTIK FINDER».

### 1.2 Найменування організації-замовника та організацій-учасників робіт

Замовником виступає кафедра автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" (далі за текстом — Замовник). Адреса замовника: м. Київ, п. Перемоги 37, 18 корпус ФІОТ.

Розробник сервісу — студент групи ІС-62 кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України "Київський політехнічний інститут ім. Ігоря Сікорського" Федько Микола Романович.

### 1.3 Перелік документів, на підставі яких створюється система

Підставою для розробки “Інформаційної системи пошуку ресторанів” є завдання на дипломне проектування, затверджене кафедрою автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи.

Автоматизовані системи.

					ДП ІС-6225. 01.000 ТЗ	Арк. 3
Змн.	рк.	докум.	дпис	Дата		

- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

#### **1.4 Планові терміни початку і закінчення роботи зі створення системи**

Плановий термін початку роботи над створенням застосування – 13 квітня 2020 року.

Плановий термін по закінченню роботи над створенням застосування – 15 травня 2020 року.

## 2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ ЗАСОБІВ

### 2.1 Призначення засобів

Даний програмний продукт призначений для того, щоб спростити пошук ресторану для користувача, шляхом формування рекомендацій. Розроблювана веб-платформа надаватиме рекомендації користувачеві незалежно від його досвіду користування платформою. Система повинна аналізувати уподобання, оцінки та досвід користувача, а також досвід інших користувачів, щоб сформувати для нього цільові пропозиції.

### 2.2 Цілі створення засобів

Цілі створення системи:

- Надати можливість отримувати кращі пропозиції від ресторанів тим користувачам, що ще не мають досвіду користування платформою;
- Надати можливість користувачам отримувати релевантні рекомендації згідно з уподобаннями та попереднім досвідом;
- Спростити життя користувачів, скоротивши час на пошуки ресторану.

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- Розробити алгоритм для формування рекомендацій на основі рейтингу найпопулярніших пропозицій;
- Розробити алгоритм для формування персоналізованих рекомендацій;
- Розробити зручний та зрозумілий інтерфейс для користування системою;

					ДП ІС-6225. 01.000 ТЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Інформаційною системою пошуку ресторанів користувач зможе користуватись без попередньої реєстрації у системі, назовемо незареєстрованого користувача гостем. Гість веб-платформи «RESTIK FINDER» матиме змогу скористуватись пошуком ресторану, страви чи кухні або ж отримати рекомендації, які будуть сформовані на основі найпопулярніших пропозицій системи. Гість також має можливість зареєструватися в системі. Зареєстрований користувач має більше можливостей, а саме: користувач зможе вказати свої уподобання щодо страв, також зможе виставляти оцінки та залишати відгуки про відвідані ресторани, що у майбутньому позитивно складеться в плані формування персональних рекомендацій для користувача. Тобто система надаватиме зареєстрованому користувачеві більш цільові пропозиції.

Об'єктом автоматизації є задача побудови пошуково-рекомендаційної системи.

					ДП ІС-6225.01.000 ТЗ	Арк.А
						6
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

## 4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 4.1 Вимоги до функціональних характеристик

Інформаційна система пошуку ресторанів являє собою веб–платформу, яка передбачає користування для 3 типів користувачів: адміністратора системи, зареєстрованого користувача та не зареєстрованого користувача, тобто гостя. Різниця полягає у функціональних можливостях та рівнях доступу для користувачів.

Акторами системи є:

- CUSTOMER – зареєстрований користувач системи;
- GUEST – гість системи
- ADMIN – адміністратор системи.

Функції, які будуть виконуватись акторами:

CUSTOMER:

- Додавання / видалення уподобань та переваг;
- Перегляд рейтингу ресторанів;
- Перегляд персональних рекомендацій;
- Публікація оцінки та відгуку для страви / ресторану;
- Пошук страви / ресторану / кухні у системі;
- Фільтрація / сортування рейтингів, рекомендацій та результатів пошуку за ціною, оцінкою, назвою чи місцезнаходженням.

GUEST:

- Реєстрація в системі;
- Перегляд рейтингу ресторанів;
- Пошук страви / ресторану / кухні у системі;
- Фільтрація / сортування рейтингів, рекомендацій та результатів пошуку за ціною, оцінкою, назвою чи місцезнаходженням.

					ДП ІС-6225.01.000 ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		



## ADMIN:

- Перегляд списку зареєстрованих користувачів;
- Надання доступу адміністратора для користувача;
- Видалення користувача;
- Додавання / видалення / редагування даних про страву, кухню чи ресторан.

**4.2 Вимоги до надійності**

**4.2.1. Передбачити контроль введення інформації.**

**4.2.2. Передбачити захист від некоректних дій користувача.**

**4.2.3. Забезпечити цілісність інформації в базі даних.**

**4.3 Вимоги до складу і параметрів технічних засобів**

Для правильної роботи розробленого застосування до складу технічних засобів повинен входити комп'ютер з операційною системою Windows 7 або вищої версії та зі встановленими браузерами, на яких буде проводитись тестування, та Visual Studio 2017 чи вище. Також комп'ютер має бути підключений до мережі Інтернет.

Вимоги до браузерів наведені в таблиці 4.1.

Таблиця 4.1

Найменування браузера	Вимоги до браузера
Google Chrome	Версія не нижче 12.0.712.0
Firefox	Версія не нижче 60
Internet Explorer	Версія 11
Opera	Версія не нижче 62

## 5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Основні етапи виконання робіт з розробки наведені в таблиці 5.1.

Таблиця 5.1

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту
1.	<i>Вивчення рекомендованої літератури</i>	10.03.2020 р.
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	15.03.2020 р.
3.	<i>Постановка та формалізація задачі</i>	28.03.2020 р.
4.	<i>Розробка інформаційного забезпечення</i>	7.05.2020 р.
5.	<i>Алгоритмізація задачі</i>	22.03.2020 р.
6.	<i>Обґрунтування використовуваних технічних засобів</i>	30.03.2020 р.
7.	<i>Розробка програмного забезпечення</i>	10.04.2020 р.
8.	<i>Налагодження програми</i>	17.04.2020 р.
9.	<i>Виконання графічних документів</i>	23.04.2020 р.
10.	<i>Оформлення пояснювальної записки</i>	01.05.2020 р.
11.	<i>Подання ДП на попередній захист</i>	15.05.2020 р.
12.	<i>Подання ДП на основний захист</i>	01.06.2020 р.
13.	<i>Подання ДП рецензенту</i>	02.06.2020 р.

## 6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

### 6.1 Види випробувань

Для перевірки коректності роботи програмного продукту буде проведено функціональне тестування. В ході тестування буде виконано перевірку всіх функціональних характеристик веб-застосунку. Також, система буде перевірена на відмовостійкість шляхом виконання некоректних дій користувачем.

Окрім функціонального тестування буде проведена перевірка безпеки системи в цілому.

					ДП ІС-6225.01.000 ТЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:  
Попенко Володимир Дмитрович

ID перевірки:  
1003947963

Дата перевірки:  
11.06.2020 01:43:23 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
12.06.2020 01:43:13 EEST

ID користувача:  
77149

Назва документу: Fedko\_is62\_2

ID файлу: 1003963112 Кількість сторінок: 55 Кількість слів: 8125 Кількість символів: 70122 Розмір файлу: 116.44 KB

## 14.8% Схожість

Найбільша схожість: 3.13% з джерело бібліотеки. ID файлу: 1000044118

7.37% Схожість з Інтернет джерелами

169

Page 57

14.8% Текстові збіги по Бібліотеці акаунту

618

Page 59

## 0.28% Цитат

Цитати

1

Page 60

Вилучення переліку посилань вимкнено

## 0% Вилучень

Вилучений текст відсутній

## Підміна символів

Не знайдено заміненних символів

**Графічний матеріал  
до дипломного проєкту**

на тему:

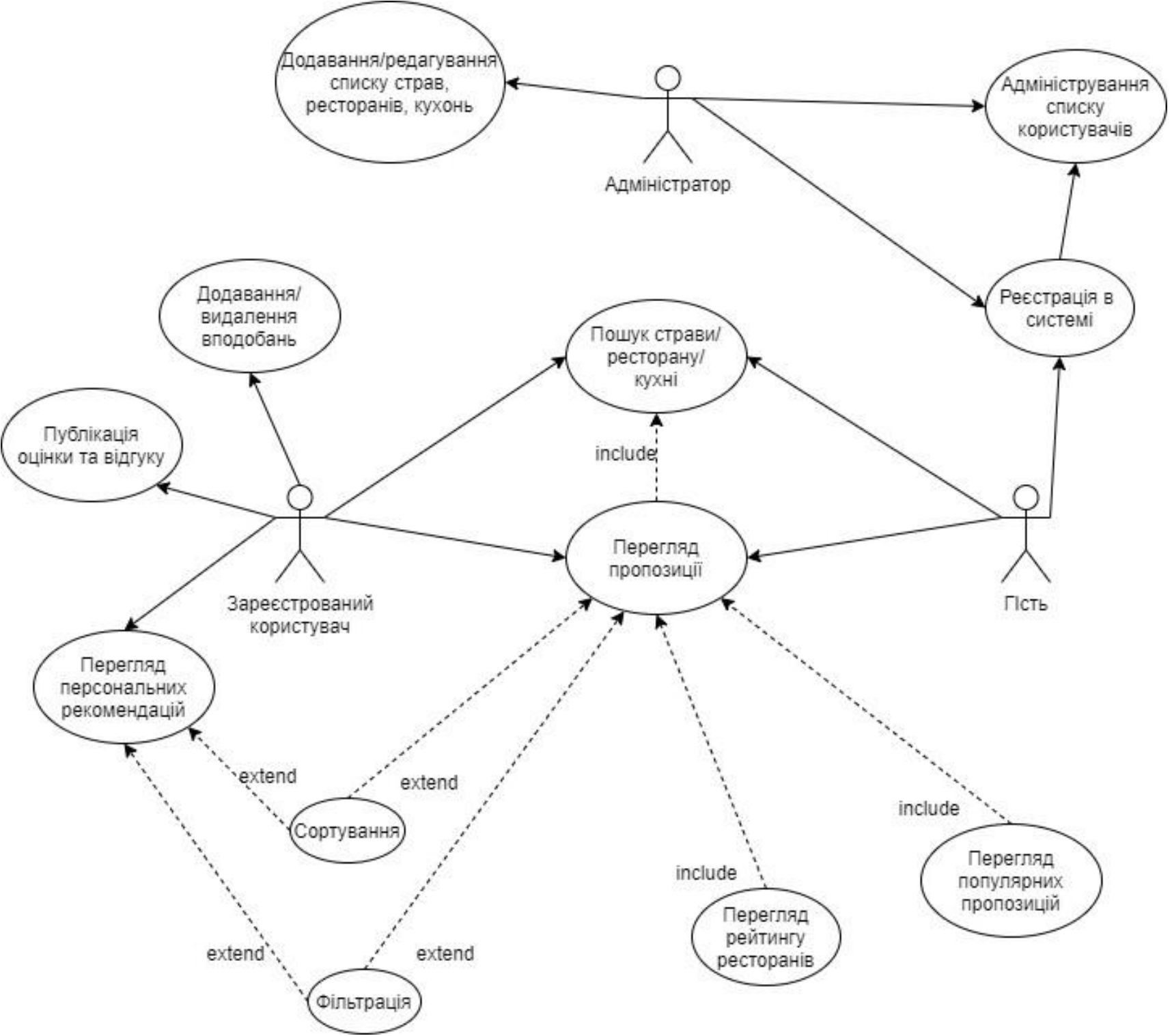
Інформаційна система пошуку ресторанів

---

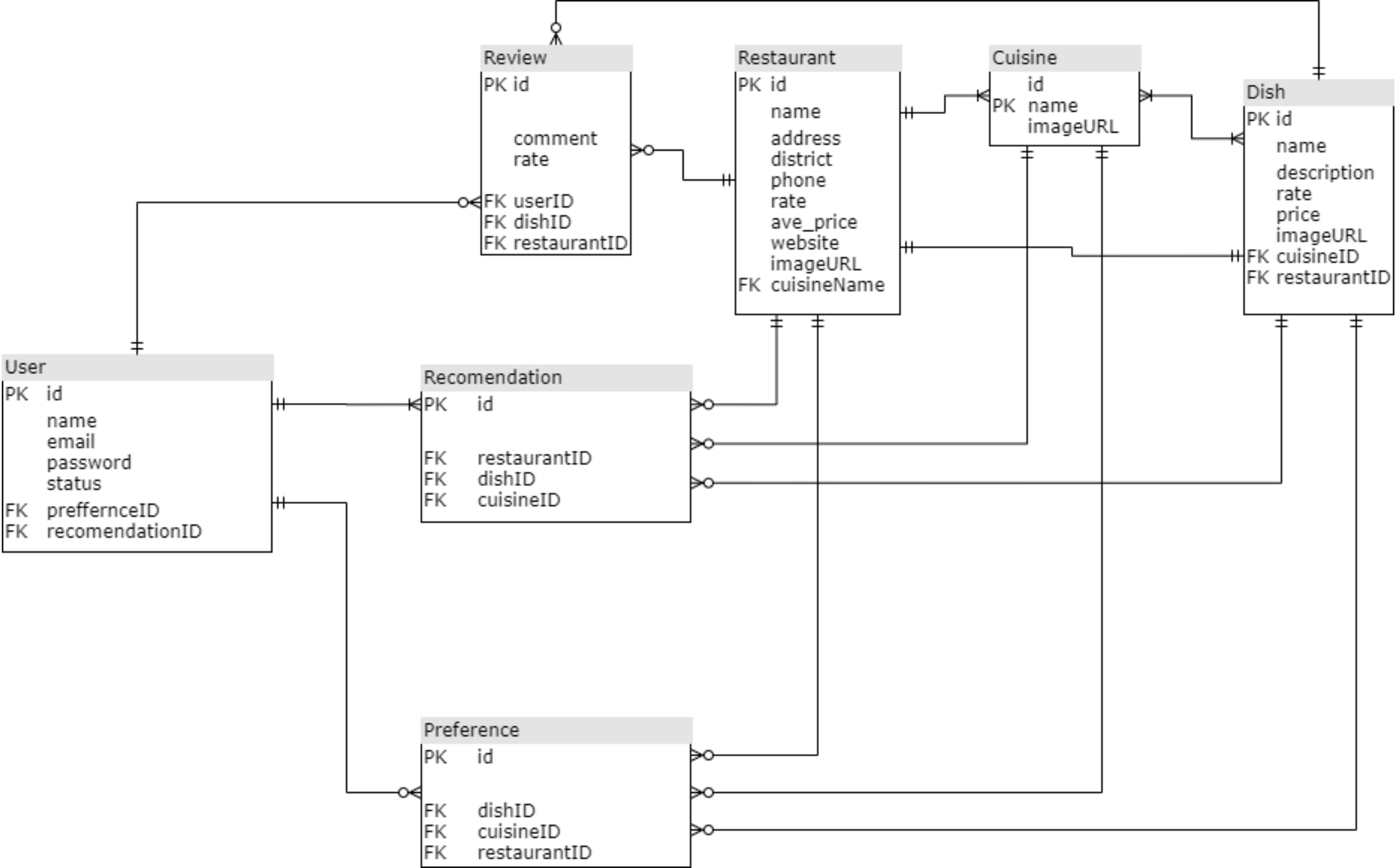
---

---

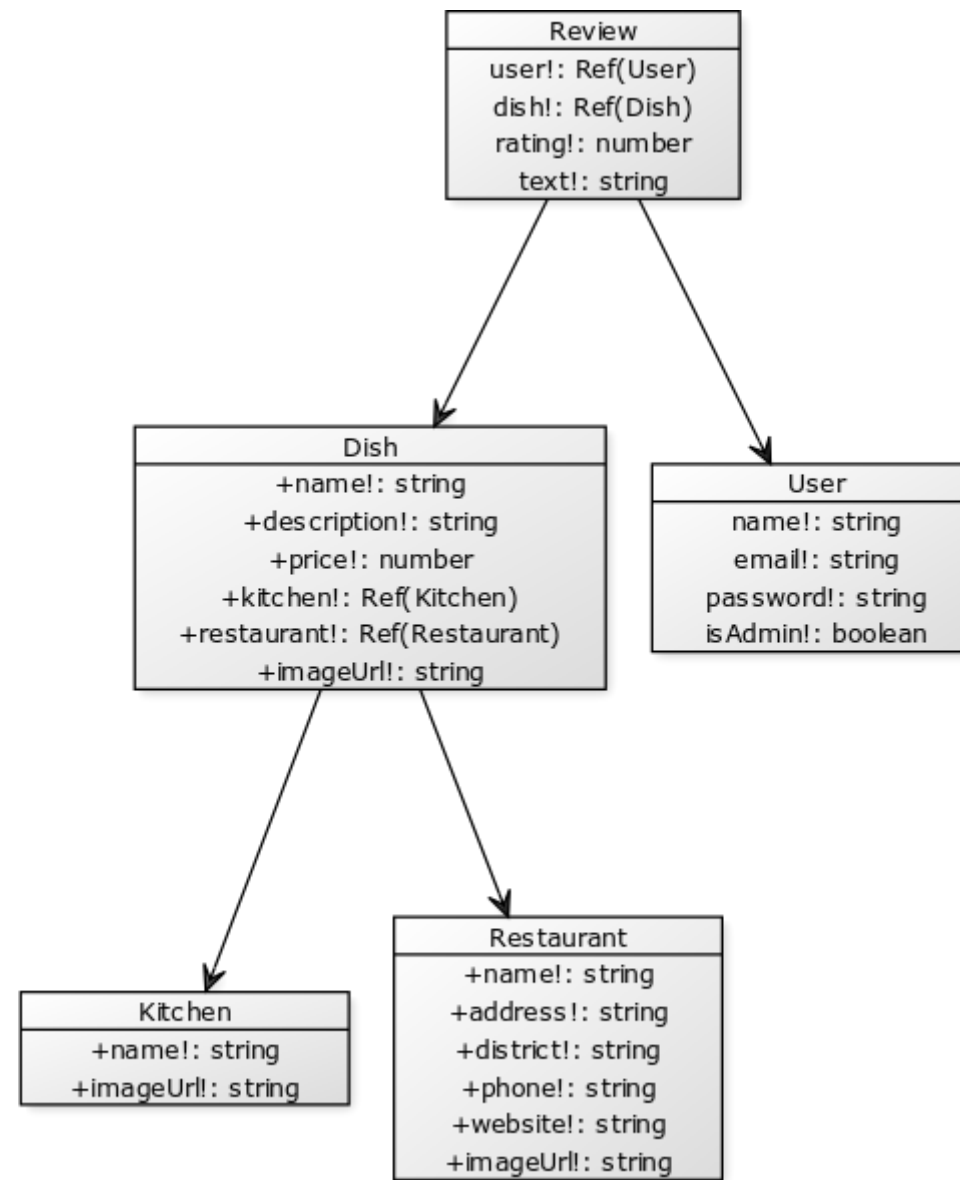
Київ – 2020 року



					ДП 6225.02.000 ССВ			
					Схема структурна варіантів використання			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Федько М.Р.			Інформаційна система пошуку ресторанів			
Перевірив		Ковтунець О.В.						
Консульт.					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			
Н. кон.		Новінський В.П.						
Затвердив		Ковтунець О.В.						
					Літера		Маса	Масштаб
					Аркуш 1		Аркушів 1	



					ДП 6225.03.000 СБД						



CREATED WITH YUML

					Схема структурна класів програмного забезпечення	Літера	Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив	Федько М.Р.							
Перевірів	Ковтунець О.В.				Інформаційна система пошуку ресторанів	Аркуш 1		Аркушів 1
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Н. кон.	Новінський В.П.							
Затвердив	Ковтунець О.В.							



cmp Components

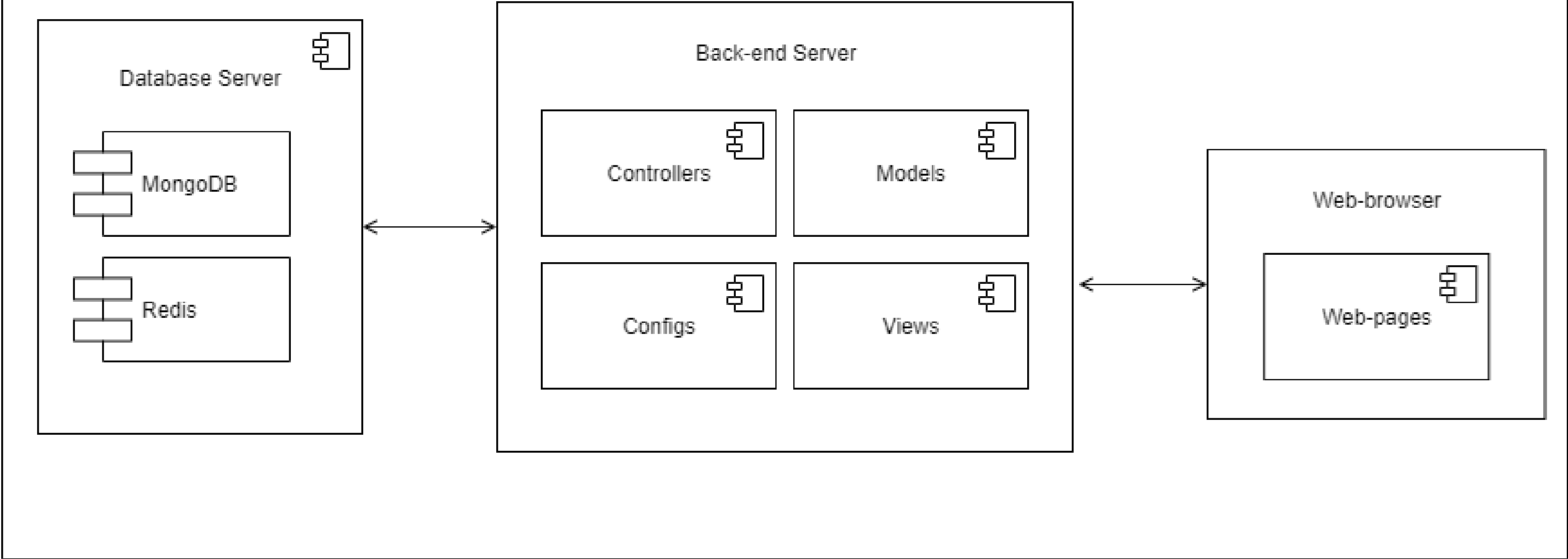
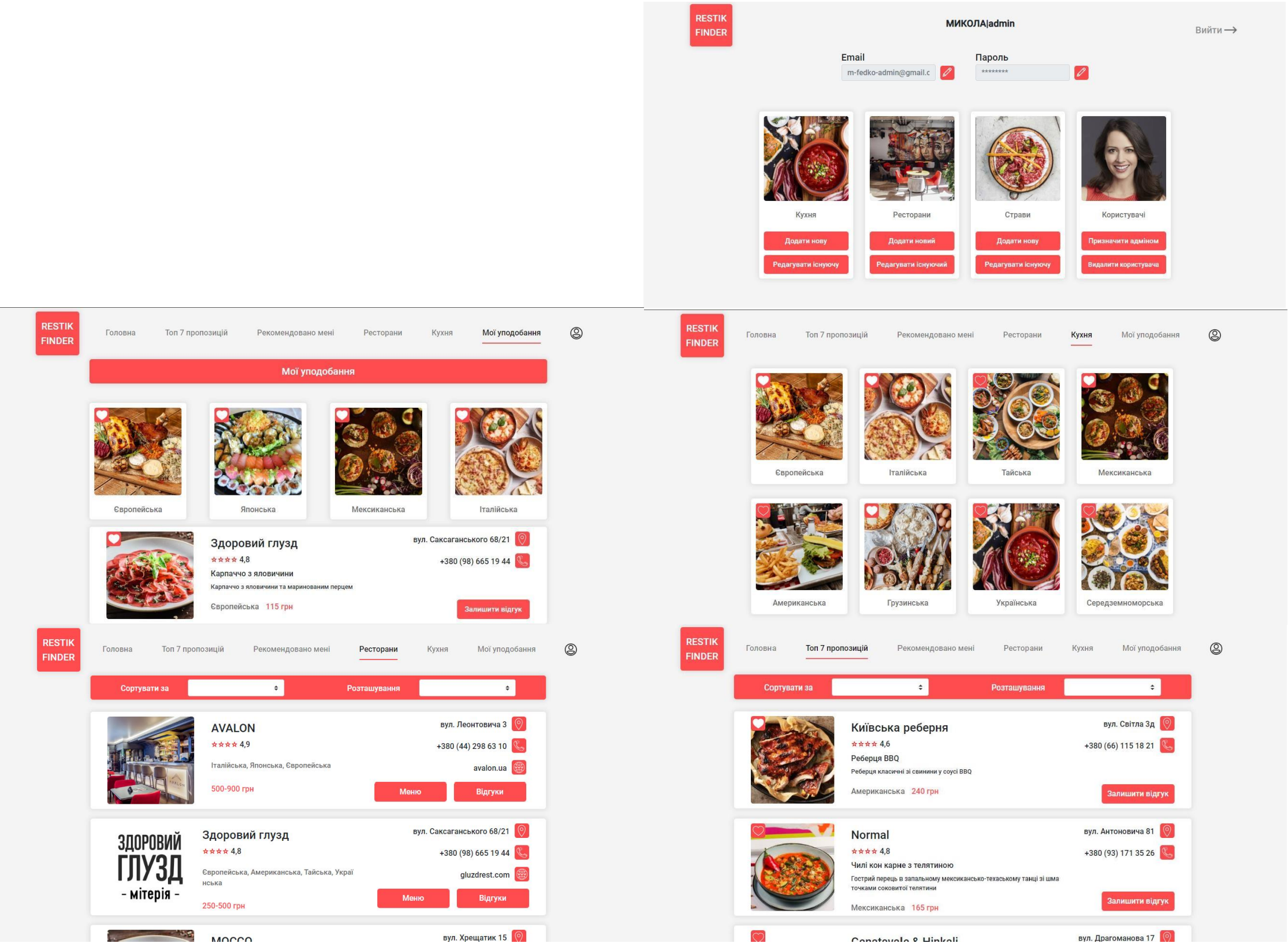
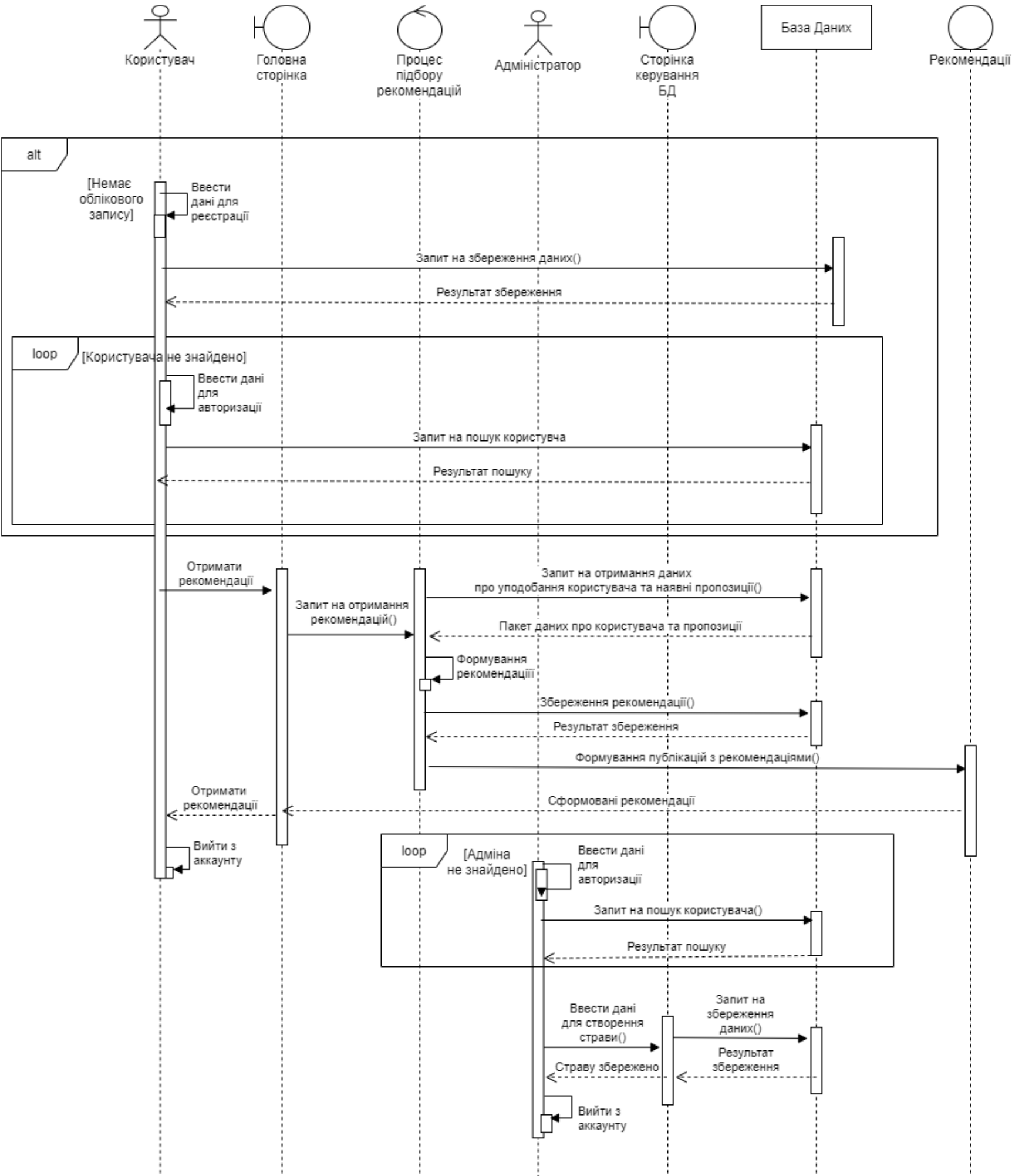


					Схема структурна компонентів програмного забезпечення	Літера		Маса	Масштаб
Зм.	Арк.	№ документа	Підпис	Дата					
Розробив		Федько М.Р.							
Перевірив		Ковтунець О.В.			Інформаційна система пошуку ресторанів	Аркуш 1		Аркушів 1	
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			
Н. кон.		Новінський В.П.							
Затвердив		Ковтунець О.В.							



Зм.	Арк.	№ документа	Підпис	Дата	Креслення вигляду екранних форм	Літера		Маса	Масштаб
Розробив	Федько М.Р.				Інформаційна система пошуку ресторанів	Аркуш 1		Аркушів 1	
Перевірив	Ковтунець О.В.								
Консульт.									
Н. кон.	Новінський В.П.								
Затвердив	Ковтунець О.В.								



					ДП ІС-6225.05.000 ССП						
					Схема структурна послідовності	Лит.		Маса		Масштаб	
Зм.	Арк.	докум.	Підп.	Дата							
Розроб.	Федько М.Р.										
Перев.	Ковтунець О.В.										
Т. Кон.						Аркуш 1		Аркушів 1			
					Інформаційна система пошуку ресторанів	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-62					
Н. Кон.	Новінський В.П.										
Затв.		Ковтунець О.В.									